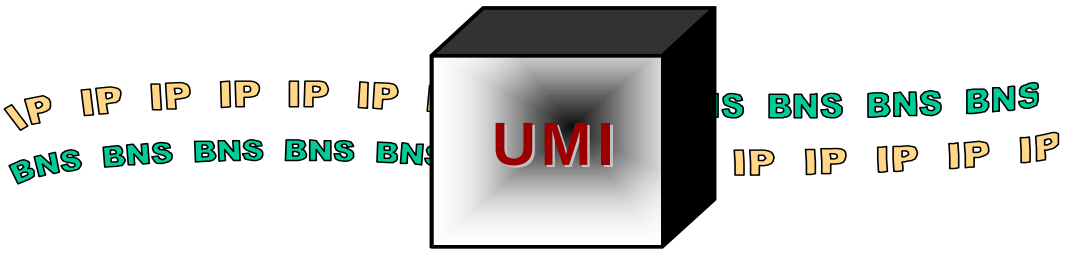
UNIVERSAL MEDIATION INTERFACE (UMI) MODULE USER'S MANUAL

Release 28.x



Manufacture & Distribution:

102 SW Orange Blossom
Lake City, Florida
32025-1613
phone: 386-754-5700
email: sales@trdcusa.com
http://www.trdcusa.com

http://www.datatekcorp.com

CONTENTS

1 INTRODUCTION 6

1.1 FEATURE SUMMARY 6

1.1.1 504 Virtual Sessions per UMI 6

1.1.2 Seamless Operation in Either Call Set-up Direction 6

1.1.3 Closed User Groups (CUGs) 6

1.1.4 Hunt Groups 6

1.1.5 Domain Name Server (DNS) capability 6

1.1.6 Peer to Peer Encryption 6

1.1.7 TACACS+ RADIUS LOGIN Support 7

1.1.8 Virtual PAD Support 7

2 PHYSICAL DESCRIPTION 8

2.1 FACEPLATE 8

2.1.1 Light Emitting Diodes (LED) 8

2.1.2 Mode Switch 8

2.1.3 Reset Button 8

2.2 High Performance I/O Board 9

2.2.1 Serial 9

2.2.2 Console 10

2.2.3 10/100 BaseT and Fiber LAN 10

2.2.4 DSU 10

2.2.5 Variants 10

2.3 Standard Performance I/O Board 11

2.3.1 Console 11

2.3.2 LAN 11

3 INSTALLATION 12

3.1 INSERTING THE I/O BOARD 12

3.2 REMOVING THE I/O BOARD 12

3.3 INSERTING THE UMI MODULE 12

3.4 REMOVING THE UMI MODULE 12

3.5 CABLING 13

3.5.1 Cabling the I/O Console Port 14

3.5.2 Cabling the CEY5 LAN Port 15
3.6 Field Upgrade and Software Registration 16

4 CONFIGURATION 19
4.1 OVERVIEW 19
4.2 BASE CONFIGURATION 20
4.3 OPERATIONAL CONFIGURATION 21
4.3.1 BNS-to-IP Calling 21
4.3.2 IP-to-BNS Calling 26
4.4 CLOSED USER GROUPS 30
4.4.1 IP-to-BNS Calling 30
4.4.2 Telnet Console 31
4.4.3 SNMP Interface 31
4.5 UMI CONFIGURATOR 31

5 UMI MODULE COMMAND REFERENCE 32
5.1 LOGIN 32
5.2 LOGOUT 32
5.3 CHGPASS 32
5.4 LOCAL 32
5.5 GATEWAY 33
5.6 DOMAIN NAME SERVER 33
5.7 TACACS+ RADIUS Servers 33
5.8 HELP 34
5.9 VERSION 34
5.10 REBOOT 34
5.11 REMOVE MODULE 34
5.12 RESTORE MODULE 35
5.13 CLEAR MEASUREMENTS 35
5.14 DISPLAY MEASUREMENTS 35
5.15 VERIFY 35
5.16 HOST NAME ADMINISTRATION 35
5.17 Mnemonic PDD ADMINISTRATION 36
5.18 SNMP 36
5.19 INSTALL (Software Registration) 37
5.20 RSTPASS (Resetting the Password) 38
5.21 CONSOLE TIMEOUT 38

5.22 VIRTUAL PORT ADMINISTRATION 39
5.23 VIRTUAL PAD ADMINISTRATION 42
5.24 VIRTUAL PAD SNOOPER 44
5.25 CLOSED USER GROUP (CUG) ADMINISTRATION 45
5.26 DISPLAY CONNECTIONS 45
5.27 DISCONNECT PORT 45
5.28 PING 45
5.29 TraceRoute 46
5.30 Label 46
5.31 Administrative Password 47
5.32 CONSOLE Administration 47
5.33 HPIO Administration 47
5.34 ADMINISTER SECURITY BANNER 48
5.35 DATA-BASE RESET 48

6 UMI SNMP AGENT 49
6.1 SNMP Version 1 Commands 49
6.2 UMI SNMP MIB Variable Database 49

7 SUPPORTED TRAPS 51

8 APPENDIX A: UMI MODULE MEASUREMENTS 52

9 APPENDIX B: UMI VIRTUAL PORT MEASUREMENTS 54

10 APPENDIX C: ALARMS 55
10.1 Major Alarms 55
10.2 Minor Alarms 55
10.3 Info Alarms 55

11 APPENDIX D: USING STARKEEPER TO CONFIGURE THE UMI 56
11.1 PRE-CONFIGURATION ACTIVITIES 56
11.1.1 Package Installation 56
11.1.2 Develop Configuration Information 56
11.1.3 Perform BNS Activities 58
11.1.4 Perform UMI Activities 58
11.2 USING THE UMI CONFIGURATOR 58
11.2.1 UMI Configurator Menu of Operations 58

11.2.2 Running the UMI Configurator 59
11.3 POST-CONFIGURATION ACTIVITIES 59

12 APPENDIX E: UMI HUNT GROUP DEMONSTRATION 60

13 APPENDIX F: UMI CLOSED USER GROUP DEMONSTRATION 61

14 Appendix G: Virtual PAD Example 63

15 Hardware Warranty 64

16 Software End-User License Agreement 64
16.1 Software License 64
16.2 Intellectual Property Rights 64
16.3 Software Support 64
16.4 Export Restrictions 64
16.5 Limited Warranty 65
16.6 No Other Warranties 65
16.7 Limitation of Liability 65
16.8 Special Provisions 65

17 Sales & Distribution 66

18 Author 66

1 INTRODUCTION

Connectivity between an IP network and a BNS (refers to both BNS-2000 and BNS-2000 VCS) network is accomplished using the **Universal Mediation Interface** Module ***(UMI)***. As a state-of-the-art "solid state" module (no disks or fans) which resides in a BNS node, the ***UMI*** is both a replacement and enhancement of the LCS60 product.

The ***UMI*** allows both synchronous and asynchronous endpoints connected to a BNS network to access endpoints on an IP network. Similarly, endpoints on an IP network can access both synchronous and asynchronous endpoints on a BNS network.

The ***UMI*** can be located anywhere in the BNS network, thus simplifying configuration, administration and maintenance without affecting operation or connectivity.

1.1 FEATURE SUMMARY

The ***UMI*** provides a complete mediation interface (i.e., gateway) for BNS and IP protocols. The basic features are:

1.1.1 504 Virtual Sessions per UMI

The ***UMI*** operates like a "virtual SAM504", in effect mapping 504 SAM ports within a BNS network to 504 "virtual ports" residing in the IP network, and vice versa.

1.1.2 Seamless Operation in Either Call Set-up Direction

The ***UMI*** provides the terminal/PC user on either network with a "terminal server" type interface where a connection setup to a user/device on the other network can be made to appear to the calling user as if the destination endpoint is on the same network.

1.1.3 Closed User Groups (CUGs)

The ***UMI*** supports Closed User Groups (CUGs) to restrict callers to specified destinations. This is an important feature for protecting sensitive endpoints in a corporate wide network without the burden of special "security servers".

1.1.4 Hunt Groups

A Hunt Group is a set of ***UMI*** "virtual ports", which are arranged to receive calls from the IP network at a common address.

1.1.5 Domain Name Server (DNS) capability

The ***UMI*** can maintain a set of mnemonic host names, analogous to the /etc/hosts file on both UNIX and Microsoft Windows platforms. This allows the ***UMI*** to perform a translation between a user-provided name and its associated IP address and TCP port number, for BNS-to-IP calls. The use of a mnemonic name is optional; the ***UMI*** will always accept an IP address in its base form.

Additionally, the ***UMI*** allows for the definition of an external DNS to be used for translation of mnemonic addresses not defined in the internal host table. Three DNS name server IP addresses are supported by the **UMI**.

1.1.6 Peer to Peer Encryption

The **UMI** can encrypt user data on the IP network. This prevents unauthorized access to sensitive information. Encryption can be enabled on a per session basis.

1.1.7 TACACS+ RADIUS LOGIN Support

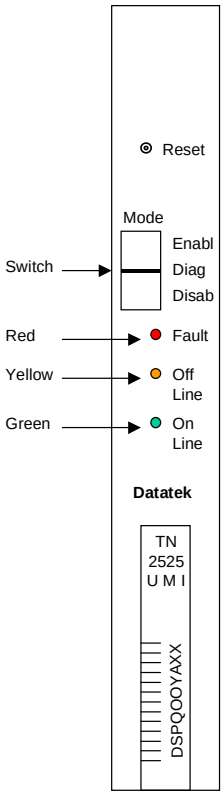
The **UMI** supports up to two TACACS+ RADIUS servers for login authentication. These are a primary, and a secondary, although each is individually enabled. The TACACS+ support is for either encrypted, or clear authorization. Encryption keys may contain spaces.

1.1.8 Virtual PAD Support

The **UMI** allows individual virtual ports to be defined a PAD functions. This allows IP resident operations sytems (e.g. REACT) to communicate with network elements resident on DK/BNS X25P and other modules. The PAD implementation is similar to that implemented on the DT-xx8x devices.

At the current time, the Virtual Pad feature is only available to a UMI that is administered in the DKCC as a SAM504.

2 PHYSICAL DESCRIPTION



2.1 FACEPLATE

2.1.1 Light Emitting Diodes (LED)

The lights on the module faceplate are green, yellow, and red. They indicate on-line, off-line, and fault states respectively. When the module circuitry detects an on-board fault, the red LED *(fault)* is lit.

2.1.2 Mode Switch

The Mode Switch supports three positions: Enabl, Diag and Disab. The Mode switch must be in the Enabl position for the ***UMI*** to function properly.

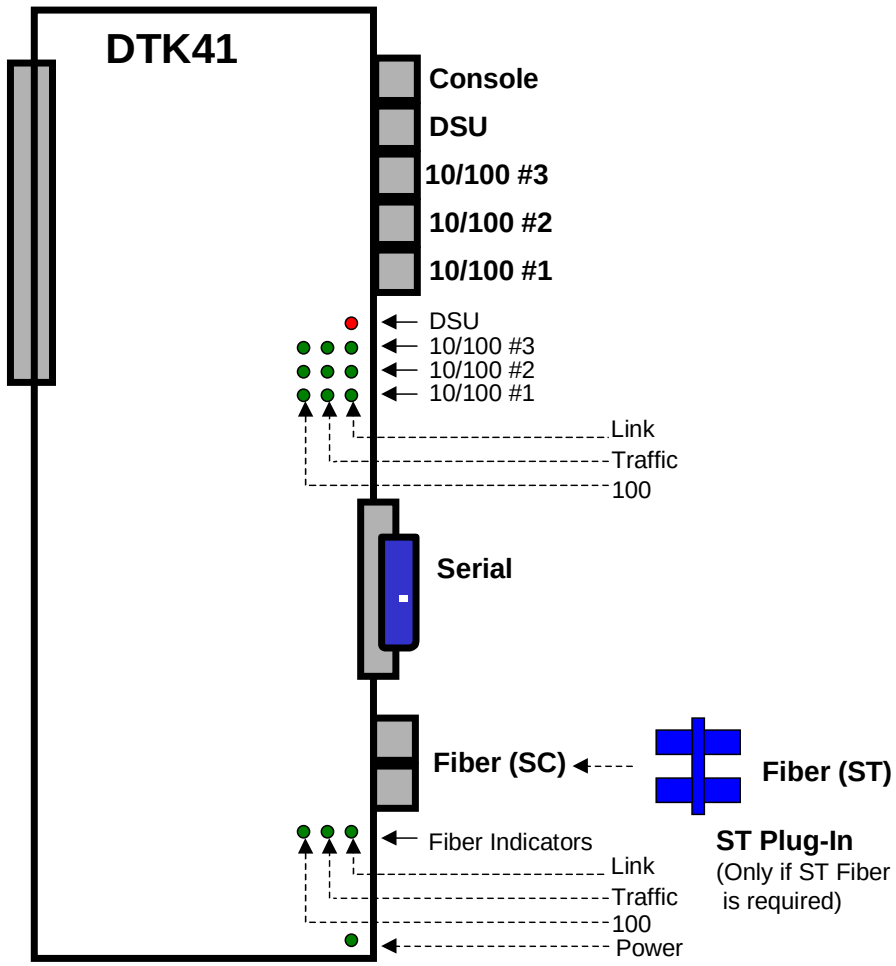
2.1.3 Reset Button

When the Reset button is pressed, the module buffers and registers are cleared, and the module application program is restarted. The module is taken out of service, and all connections are terminated.

2.2 High Performance I/O Board

The UMI[1] mates with the DTK41 High Performance I/O (HPIO) board in support of the UTM connectivity options. The DTK41 I/O board contains all the necessary connectors the UTM requires for currently available Console and LAN connections.

Note: Cables and adapters are available



2.2.1 Serial

The UMI does not use the serial interface of the DTK41 at this time. The DTK41 Serial interface is software configurable. Options are as an RS-232C DTE for serial rates up to 64Kbps, or as a V.35 DTE for serial rates up to E1.

[1] A Series 1:2 UMI is required for the DTK-41 (or any other I/O board which requires power) since the original UMI did not provide any power whatsoever to the I/O module, and the original UMI did not provide a management interface.

2.2.2 Console

The UMI Console interface may be used for console activities and the initial configuration. It assumes the connected device is configured as 9600 baud, 8 bits and no parity.

2.2.3 10/100 BaseT and Fiber LAN

The DTK41 LAN and Fiber ports are used for IP network connectivity. The UMI supports Internet peer level protocols *(e.g. IP, TCP, UDP, ICMP and SNMP)*. All of the 10/100 ports are fully switched, not bridged. The capacity of the switch is over 1Gbps. The 10/100 and Fiber ports are managed from the UMI console. Each 10/100 port and the Fiber port may be enabled or disabled individually. Alarms are generated if a link is established, or if a link is lost.

An advanced feature DTK41 is that no crossover cable is ever required on the 10/100 ports. The 10/100 ports will automatically correct for the cabling mismatch.

The 10/100 ports will self-configure to match the speed of the link (10 Mbps or 100 Mbps), and the Duplex (Full or Half). No configuration is required.

The Fiber interface connects to the industry standard SC cabling. If ST cabling is used, the DTK41 ST plug-in module is used. No adapter cables are required.

2.2.4 DSU

The UMI does not utilize the DSU interface at this time. The DTK41 DSU *(4-wire)* interface is software configurable for T1 or E1 rates. A value of T1 is used for domestic 1.544 MHz interfaces with 193 Bit Superframes. A value of E1 is used for European 2.048 MHz interfaces with 256 Bit Superframes. The DSU functionality is built in. This interface may be used for connectivity to TDM, Frame Relay, and ATM networks.

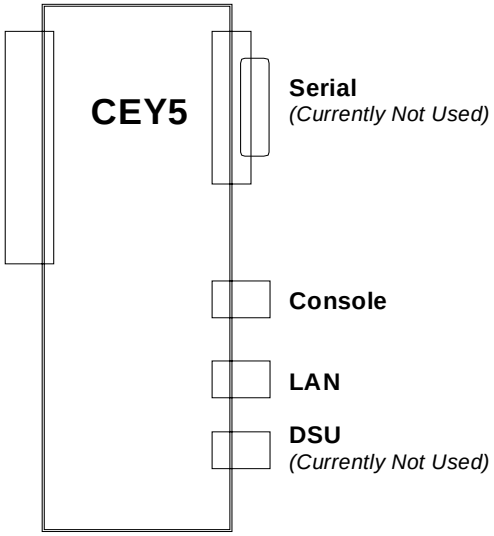
2.2.5 Variants

The **DTK41** high performance I/O board is available with two different mounting brackets. The **DTK41N** incorporates a mounting bracket specifically designed to be installed in a BNS node, a Datakit VCS node, or an MPC. The **DTK41S** has a mounting bracket specifically designed to be placed in a SAM504, SAM64, or SAM128. It is anticipated that the UMI shall use only the **DTK41N** variant of the **DTK41**.

2.3 Standard Performance I/O Board

The ***UMI*** mates with the CEY5 I/O board in support of the ***UMI*** connectivity options. The CEY5 I/O board contains all the necessary connectors the ***UMI*** requires for currently available Console and LAN connections.

Cables and adapters are available.



2.3.1 Console

This interface requires a standard RJ45 terminated, twisted pair, data cable. It connects as a data terminating equipment (DTE) to an asynchronous device and uses RS-232C signaling. Connection to the ***UMI*** console is required for ***UMI*** "Module-Based" Administration (described later in this document) or StarKeeper® II NMS alarm collection[2]. Otherwise, the console can be disconnected during normal operation.

2.3.2 LAN

This interface requires a standard RJ45 terminated Category 5, twisted pair, data cable. It connects to a 10BaseT hub or Router on the local LAN segment.

[2] The ***UMI*** also supports **console access** through a TCP telnet connection, which is accessed using **TCP port 1023.** This service is available only when the unit is in service.

3 INSTALLATION

UMI installation consists of:

- inserting the DTK41 or CEY5 distribution board in the backplane slot
- inserting the module in the corresponding shelf slot
- cabling console and data ports

When installing a UMI:

- Ensure protection from electromagnetic interference **(EMI).** Wear an electrostatic discharge **(ESD)** wrist strap to prevent equipment damage.
- To prevent damage to module circuitry, always insert the I/O board before inserting its corresponding module. Never remove the I/O board before removing the module.

3.1 INSERTING THE I/O BOARD

The I/O board plugs into the backplane at the rear of the shelf; it is held in place by shrouds on the backplane pinfield, and secured with two screws. Insert the I/O board before inserting its corresponding ***UMI***.

- Align the I/O board backplane connector with the backplane pinfield, and align the screw slots with the screw holes.
- Slip the backplane connector onto the pins. The board should seat easily. If seating is difficult, the board may be canted or some pins may be bent.
- Insert the screws, and tighten them securely.

3.2 REMOVING THE I/O BOARD

Remove the I/O board only for relocation, replacement, or board type confirmation.

Requirement: The Module in the slot corresponding to the I/O board must be removed first.

- Disconnect all cabling to I/O board ports, labeling the cable ends if appropriate.
- Remove the screws holding the I/O board in place.
- Carefully rock the board as you pull it out.

3.3 INSERTING THE UMI MODULE

Requirement: The I/O board for the module must be installed in its corresponding slot on the backplane at the rear of the shelf first.

- Set the mode switch on the module faceplate to Disab.
- With the module latch extended, carefully push the module all the way into the slot. The backplane pins slip into the module receptacle.
- Close the latch to lock the module into position.
- Move the mode switch on the module faceplate to Enabl.

3.4 REMOVING THE UMI MODULE

You can remove and replace a ***UMI*** in an operating node without damaging the module itself, or without disrupting calls on other modules. Only the calls on the ***UMI*** being removed are disrupted.

Requirement: I/O board for the module must still be in its corresponding slot on the backplane at the rear of the shelf.

- If the mode switch is in the Enabl position, move it to Disab.
- Open the latch on the module faceplate.
- Pull the module straight out of the slot.

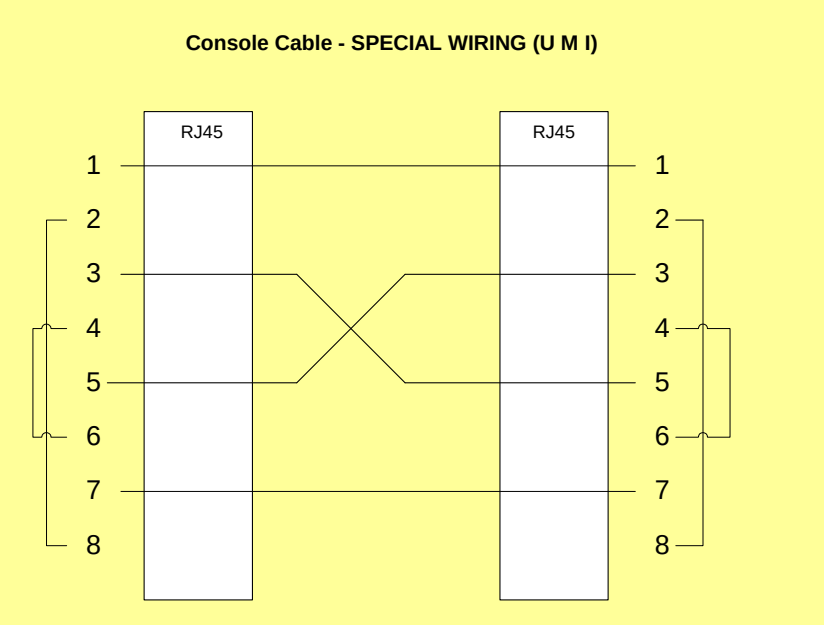
3.5 CABLING

This section provides information on how to cable the ***UMI*** console and data ports with the CEY5 I/O board. Consult the following table for ordering information regarding all of the cabling options shown in this section.

Cable / Adapter	Description	ED5P055-31 Group Number or Comcode
258B adapter	50-pin M to 6 8-pin mod	G(154)
Mod – DB15 adapter	8-pin mod to DB15 M	
D8AH-M adapter	25-pin M to mod socket	G(139)
D8AH-F adapter	25-pin F to mod socket	G(147)
Console (special wiring)	**8-pin mod to 8-pin mod**	**408198133**
Straight mod cable	8-pin mod to 8-pin mod	G(137), G(G)
CAT5 cable	8-pin mod to 8-pin mod (shielded)	

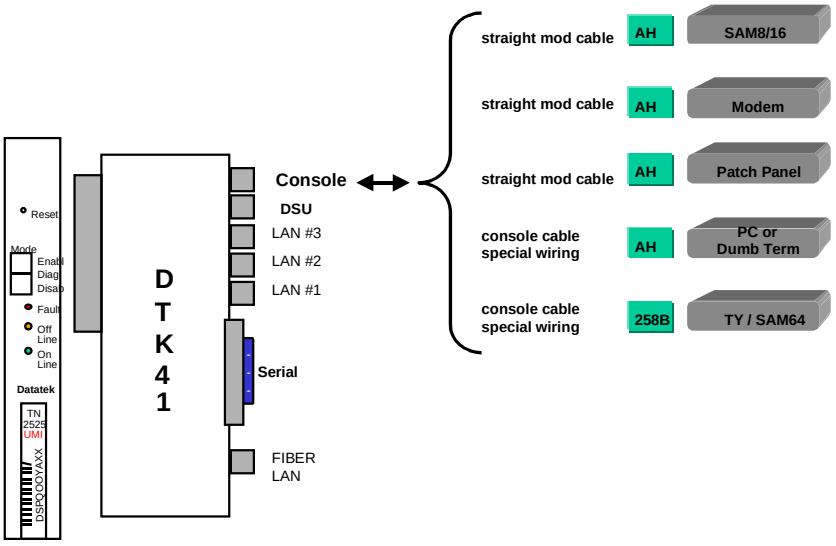
Note: The AH adapter will be used to terminate the cable and will be attached to the appropriate device. The attached device will determine the gender of the AH adapter.

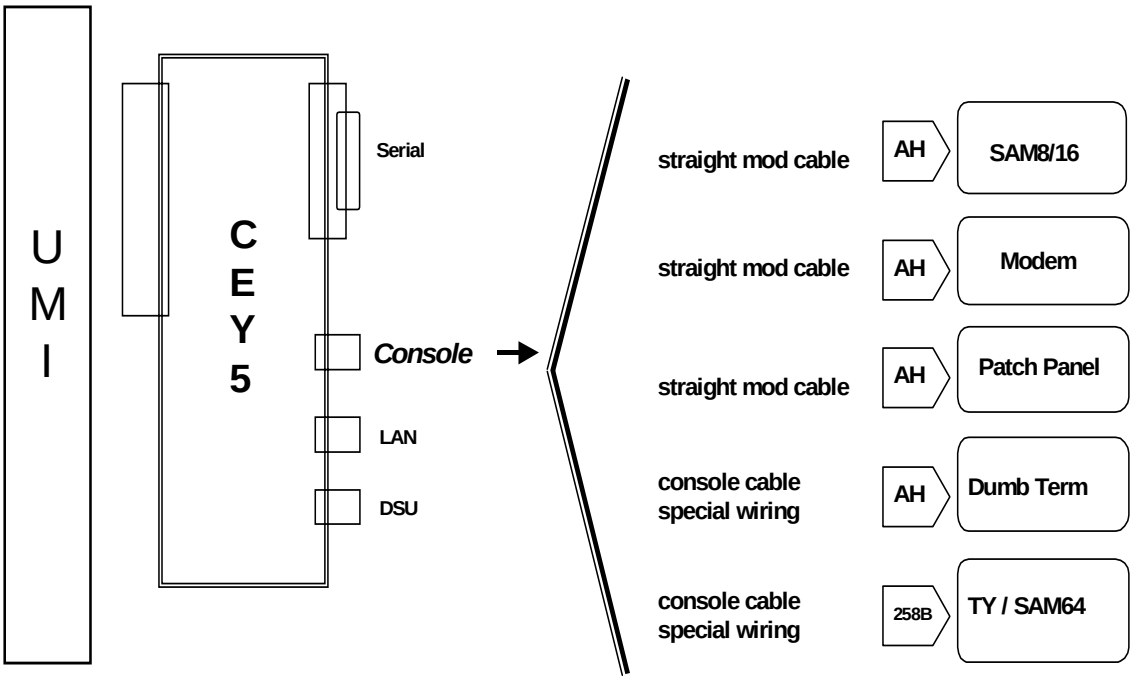
Important! A **Console** cable with "SPECIAL WIRING" can be ordered using the above table or built using the following wiring diagram.



3.5.1 Cabling the I/O Console Port

Depending upon access availability, one of the cables shown in the following diagram will be needed to set up a ***UMI*** console connection.

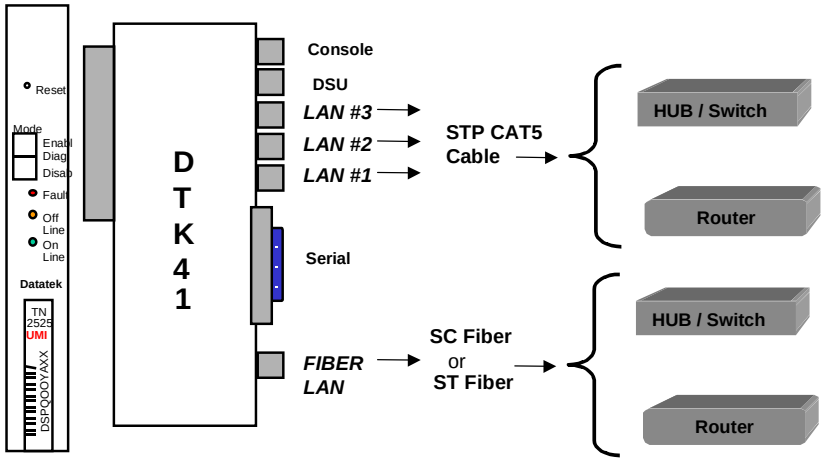


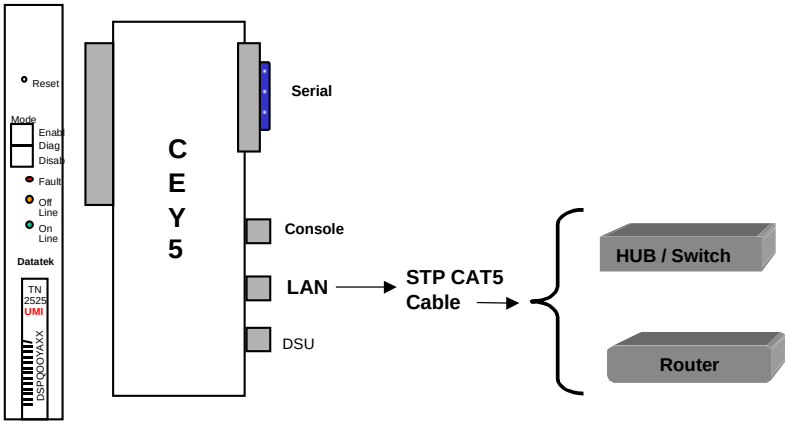


3.5.2 Cabling the CEY5 LAN Port

The following will be needed to set up a ***UMI*** data connection.

- A Shielded Twisted Pair CAT5 cable is attached to the LAN port of the I/O board and will allow for cabling to either a hub or router. The DTK41 I/O board may use a 10BaseT or 100BaseT hub or router. The CEY5 I/O board uses a 10BaseT hub or router.





3.6 Field Upgrade and Software Registration

The **UMI,** when initially delivered, will need one or more software keys to activate the software. Software keys are also required when an optional individual feature packages are added to the device. Finally, when the **UMI** is upgraded with revised software, one or more software keys are required to register the installed software and any feature packages registered for the device.

When performing an upgrade, the revised software is initially downloaded by **upgrade** into a staging area and is not active. The software then is activated by a ***reboot***. The new software will execute normally prior to registration. However, no backup, reloads, or upgrades can be performed. Module level parameters, such as the device IP address, may be changed and activated. Interface specific parameters cannot be changed.

The procedure for performing a software registration has been mechanized. Manual procedures are error prone and not recommended. They are no longer covered in this user manual.

The mechanized Software Upgrade Registration procedure allows simplified administration of one or more devices. When a quantity of devices are upgraded, manual software registration of each device has the potential of becoming increasingly tedious. The mechanized software upgrade registration process was designed to alleviate the problems associated with multiple device upgrades. It is also preferred for single device upgrades as it eliminates any potential for error.

The new software is downloaded to the **UMI** via the **upgrade** command. This may be performed for one or more devices. The "-r" option to the dtupgrade command will restart the device on the new software after the download completes successfully. It is highly recommended. In the alternative, the device may be downloaded without a restart and restarted at a later time during a scheduled maintenance window. Restarting the device on the new software prior to registration is required. After the restart, the devices will continue to operate normally on the new software without registration. Some operations interface functions are inhibited pending software registration. Below is an example of a typical **upgrade** invocation. Note the use of the "-r" option as it is recommended.

upgrade –v –d –r –mUMI 192.168.0.234 umi.28.1

Mechanized registration is performed in three steps. Each of which does not require user intervention.

The steps are as follows:

1. The **getinfo** utility is invoked on a file containing a list of devices to be administered. This file is called the master device list file and is typically named "dt_device.master". The master device list file may have any name and it is provided as an argument to

the **getinfo** utility. The master device list may also contain devices that do not require registration. The **getinfo** utility makes inquiry of each device in the master device list and creates a device information file named “dt_device.info” in the current directory.

2. The “dt_device.info” file is then sent via email to keys@trdcusa.com for registration processing.
3. A file name “dt_device.register” file is returned via email to be used as input in the next step. A file named “dt_device.msgs” is a text file that may be displayed or printed showing the results of the registration function.
4. The **setreg** utility is invoked and uses the “dt_device.register” file provided as an argument. If no argument is provided, the file is assumed to be in the current directory. The **setreg** utility contacts each device that requires registration and have been assigned keys. One or more keys are installed during the dialogue.
5. The “dt_device.info” file and the “dt_device.register” file are deleted as they are transient and have no further value. Neither can be reused for the purpose of registration. However, the dt_device.info file may be used for inventory reports..

The source for the registration procedure is the inventory master device list file that is created, and maintained, by the administrator using their favorite text editor.

The master device list file contains one IP address per line, with an optional TCP port, and an optional password override, to access the device. The IP address is the console *connection address*, and not necessarily the actual device IP address. Registration via the serial console is explicitly supported. Comments are allowed between addresses, and after addresses. A password override is only required if the default password of “initial” has been changed.

The master device file line format is as follows:

<IP ADDRESS> [<TCP PORT>] [-P<Password>] # Comment

An example “dt_device.master” file follows:

```
# This is a Sample master device list file “dt_device.master”.
# Note that there is one device ( Connect IP Address ) per line.
# TCP Port Override is allowed. Registration may use the serial console.
# Password Override is allowed.
# It is OK to have devices that do not need registration listed for inventory.
# Comments in this file are preceded with a pound symbol.
# Blank Lines are treated as comments.
# Basic Line Format is as follows:
```

10.0.1.80 # Device at Location 'A'

192.168.7.82 # Device at Location 'B'

192.168.7.155 50001 # Example of TCP port Override.

192.168.7.156 50001 –pcustom1 # Example of Password Override.

Once the "dt_device.master" file is prepared, it is used as an input to the **getinfo** utility.

dtgetinfo dt_device.master

This **getinfo** utility will collect information on each device in the master file. The **getinfo** utility will also make a determination if a registration is actually required. Consequently, the **getinfo** utility is also useful in performing inventory functions outside of the device registration. The output of the **getinfo** utility is a file named "dt_device.info" that is always created in the current directory.

The file "dt_device.info" is attached to an email and sent to the address keys@trdcusa.com.The registration procedure is performed and a file named "dt_device.register" is attached to return email to the original sender. A messages file named "dt_device.msgs" is also attached and may be printed as a report of the key generation function.

After receiving the "dt_device.register" file, the **setreg** utility is invoked with the relative path of the "dt_device.register" file as it's sole argument. The **setreg** utility will only contact the devices that actually need registration, and for which one or more keys were successfully generated. All of the appropriate keys, including a device key and multiple per port feature package keys, are installed by the **setreg** utility. The device is not restarted and this operation may occur during normal transport operation.

A report utility **devrep** is available. The **devrep** utility uses the "dt_device.info" file to display the inventory information. The usage is as follows:

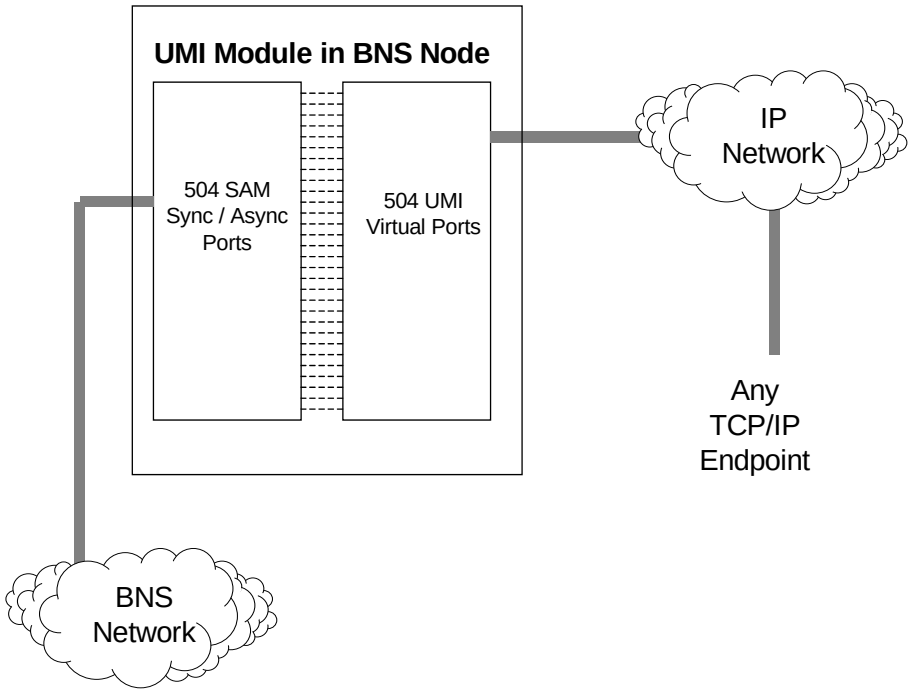
devrep [-v] dt_device.info

If the file is not specified, the dtdevrep utility attempts to use the "dt_device.info" file resident in the current directory.

4 CONFIGURATION

4.1 OVERVIEW

The following diagram is a functional representation of ***UMI*** operation, intended as an aid to understanding the configuration process.



I

In order to allow it to be deployed in any BNS node ***without requiring an upgrade of the node software***, the ***UMI*** has been designed to appear to the node controller as a T1-Trunk-connected SAM504. This requires the administrator to follow the same configuration command sequence that would be used for a SAM504 (with specific entry parameters described later in this document), using either StarKeeper® II NMS or the node's local console. Administration of additional parameters specific to the ***UMI***, referred to as "Module Based" administration, is accomplished through the ***UMI***'s console function, which is accessed via the RS-232 console port on the CEY5 I/O board or via a TCP/telnet connection from a device anywhere in the IP network. ***UMI*** "Module Based" administration can also be accomplished through StarKeeper® II NMS.

The overall configuration process can be divided into two phases:

Base Configuration – setting up the ***UMI*** for IP connectivity and console security

Operational Configuration – setting up the ***UMI*** and BNS node to enable users to make calls between the BNS and IP networks

Actual command sequences will be presented in this section to illustrate the ***UMI*** and related node configuration process. For node commands, the appropriate references are the *Data Networking Products Synchronous/Asynchronous Multiplexer Reference* and *Commands Reference*. Section 5 of this document should be used as the reference for ***UMI*** module commands.

4.2 BASE CONFIGURATION

This phase of the configuration process sets up the ***UMI*** for IP networking. This includes the ***UMI***'s IP address and subnet mask, the IP address of the gateway router, the IP address of an SNMP manager (optional), and the IP address of a domain name server (optional). It also includes setting up console-security parameters, i.e., an administrative login password and the (optional) timeout for automatic console logoff. These aspects of the configuration will rarely need to be changed once they are satisfactorily set up.

The following is an example of base configuration:

```
<UMI> login passwd=initial ↵
User is Logged in UMI
<UMI> chgpass old=initial new=secret confirm=secret ↵
Password Change Operation Succeeded.
<UMI> rm ↵
Remove
MODULE removed from service.
<UMI> local ipaddr=135.17.59.165 submask=255.255.255.0 ↵
<UMI> gateway ipaddr=135.17.59.1 ↵
<UMI> dns ipaddr1=135.17.59.6 ↵
<UMI> snmp ipaddr=135.17.59.7 ↵
<UMI> vfy mod ↵

Current Configuration:
   DK Module Type ==> TN1015 (TRK-T1)  w/Serial# 0.
   Service State ==> Out of Service.
   Local MAC Address ==> 0.96.29.2.52.22
   Local IP Address ==> 135.17.59.165
   Subnet Mask ==> 255.255.255.0
   Gateway IP Address ==> 135.17.59.1
   DNS Name Server Address ==> [#1] 135.17.59.6
   SNMP Trap Manager ==> 135.17.59.7  Port 162

<UMI> rs ↵
Restore
Module restored to service
Reboot Initiated to Make Physical Parameters effective.
```

4.3 OPERATIONAL CONFIGURATION

There is an implicit one-to-one mapping of *SAM504* boards/ports (as the node views them) to the ***UMI***'s virtual ports ("vports"), as shown in the following table. These associations are built into the ***UMI*** module software, and will need to be kept in mind for proper coordination of node and module administration procedures for operational configuration.

BNS Controller *(SAM504)* Board/Port ←------→ UMI Virtual Port mapping

***SAM504* Board/Port Numbers**	**UMI Virtual Port Numbers**
Board 1 Ports 1-32	Ports 1-32
Board 2 Ports 1-32	Ports 33-64
Board 3 Ports 1-32	Ports 65-96
Board 4 Ports 1-32	Ports 97-128
Board 5 Ports 1-32	Ports 129-160
Board 6 Ports 1-32	Ports 161-192
Board 7 Ports 1-32	Ports 193-224
Board 8 Ports 1-32	Ports 225-256
Board 9 Ports 1-32	Ports 257-288
Board 10 Ports 1-32	Ports 289-320
Board 11 Ports 1-32	Ports 321-352
Board 12 Ports 1-32	Ports 353-384
Board 13 Ports 1-32	Ports 385-416
Board 14 Ports 1-32	Ports 417-448
Board 15 Ports 1-32	Ports 449-480
Board 16 Ports 1-24	Ports 481-504

It's important to note that although a single ***UMI*** supports call setup in both directions simultaneously, each grouping of virtual ports must be configured for one direction only; i.e., there are no two-way virtual ports. In addition, virtual ports intended to be used for synchronous and asynchronous call connections must reside in different groups, since their protocol encapsulation is different.

The following subsections describe how the node and ***UMI*** are jointly configured to enable calls to be set up in each direction.

4.3.1 BNS-to-IP Calling

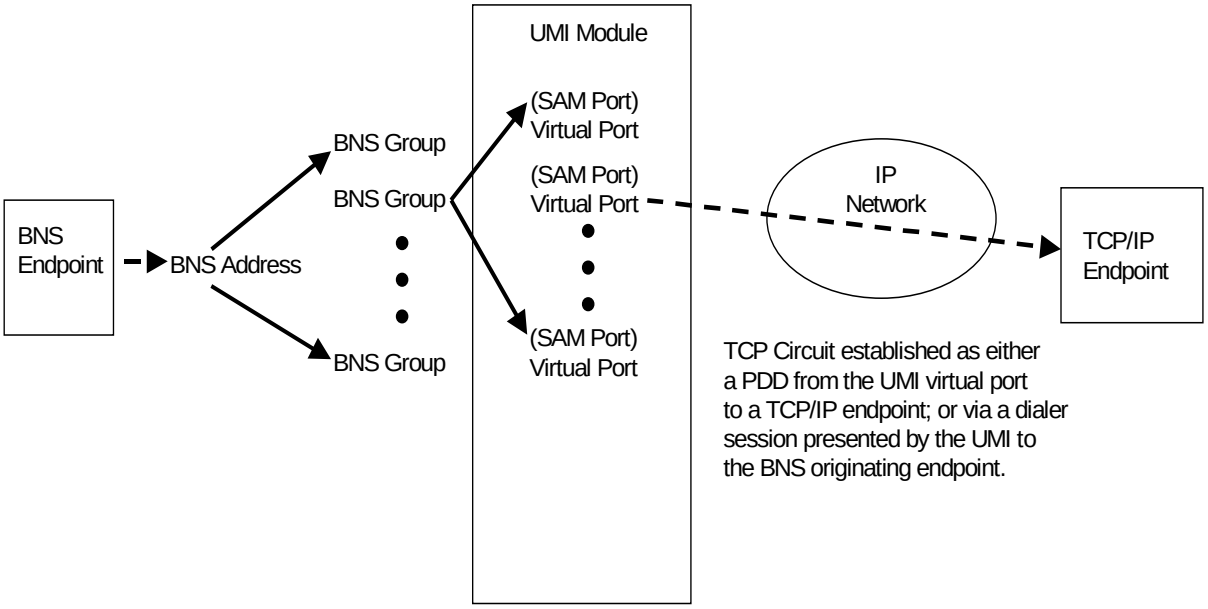
UMI virtual ports designated as ***originating,*** as part of module-based administration, are for the BNS-to-IP calling direction**.** From the node administration perspective, the corresponding SAM ports are set up as **host** service-type ports, since they receive calls from the BNS network. Predefined destination (PDD) information, in the form of a destination IP address and TCP port number, may be optionally set up on the ***UMI*** for each virtual port.

For ***UMI*** virtual ports configured as ***originating***, operation from the perspective of a user calling from somewhere in the BNS network is determined by whether or not a PDD has been specified for the virtual port chosen to originate the call. An ***originating***-type virtual port which has a PDD associated with it will have that connection automatically established at the time a BNS connection is made to the ***UMI***. If no PDD has been specified, the calling user is greeted with a **UMI Destination>** prompt each time the virtual port receives a BNS connection. The user would then enter the destination IP address plus TCP port number desired. If no TCP port number is

entered, the telnet default (23) is used. The user also has the option to enter a mnemonic host name which has been previously administered into the ***UMI***'s host table. The session is terminated when the ***UMI*** virtual port receives a disconnect from the BNS side of the call, e.g., when the calling user uses the attention sequence.

If a Domain Name Server has been defined on the ***UMI***, the calling user may also enter a fully qualified destination name *(e.g. "server.ab.company.com")* to be resolved. It is also possible to override the TCP port while still resolving the IP address. For example, the dial string "server.ab.company.com 50030" selects TCP port 50030 and then asks DNS to resolve "server.ab.company.com" to an IP address.

The following diagram illustrates the configuration elements affecting call setup in the BNS-to-IP direction.


UMI Module
BNS Endpoint
BNS Address
BNS Group
BNS Group
BNS Group
(SAM Port) Virtual Port
(SAM Port) Virtual Port
(SAM Port) Virtual Port
IP Network
TCP/IP Endpoint
TCP Circuit established as either a PDD from the UMI virtual port to a TCP/IP endpoint; or via a dialer session presented by the UMI to the BNS originating endpoint.

Referring to the above diagram, a BNS endpoint calls an address in the BNS node's configuration database which is associated with one or more ***receive*** groups. This address and the receive group(s) it is associated with would have been established using the node's **enter address** and **enter group** commands, respectively. Each receive group contains some of the ports on the "SAM504" as members. This would have been configured on the node using **enter sam** (port component), using the module address of the ***UMI*** and a series of available boards/ports from the table at the beginning of this section. When the call is established between the originating endpoint and the destination address, the appropriate receive group is selected, and, from the node's perspective, a SAM port within that group is selected to receive the call. In reality, the corresponding virtual port on the ***UMI*** receives the call. If a PDD has been established for this

virtual port, using the **vport** command (part of module administration), the call immediately proceeds to the correct IP address and TCP port in the IP network. If not, the calling user sees the **UMI Destination>** prompt.

The following is a sample ***UMI*** module administration command sequence, which will set up a group of originating virtual ports on the ***UMI***, followed by the corresponding node command sequence to enable BNS-to-IP calling:

```
<UMI> rm ↵
Remove
MODULE removed from service.

<UMI> vport 1 cnt=32 type=orig dest=135.17.59.20 dport=23 ↵

<UMI> rs ↵
Restore
Module restored to service
Reboot Initiated to Make Physical Parameters effective.

<UMI> login passwd=initial ↵
User is Logged in UMI
<UMI> vfy vport all ↵
```

(virtual ports 33 – 504 omitted)

```
    Virtual Ports   1 -  32 :
        Type ==> TCP Port w/Call Origination.
        Service State ==> Out of Service.
        PDD TCP Destination ==> 135.17.59.20  Port 23
        Protocol ==> Asynchronous.
        NULL after CR Operation ==> Transparent.
```

Note that the ***UMI*** virtual ports are not yet in service even though the module itself has been restored. They will be placed in service via node-administration commands referencing the appropriate SAM504 ports, as shown next.

The first node-administration step is to establish one or more ***receive*** groups to which SAM504 ports can be associated, using the **enter group** command. ***They should be configured with* HOST AUTOBAUD OFF**.

```
CC0> enter group ↵
GROUP [up to 8 chars]: umi-rcv ↵
TYPE [local, trunk: +(local)]: ↵
DIRECTION [originate, receive, 2way]: receive ↵
DEVICE OR HOST [up to 8 chars: +(standard)]: ↵
HOST AUTOBAUD [on, off: +(off)]: ↵
ROUND ROBIN SERVICE [per_port, per_module, none: +(none)]: ↵
```

Next, **enter address** would be used to associate a callable address (numeric and/or mnemonic) with the receive groups.

```
CC0> enter address ↵
LEVEL [network, area, exchange, local, speedcall: +(local)]: ↵
```

```
TYPE [numeric, mnemonic, both: +(mnemonic)]: ↵
MNEMONIC ADDRESS [up to 8 chars]: call-ip ↵
PAD SUPPORT [yes, no: +(no)]: ↵
DIRECTORY ENTRY [up to 30 chars double quoted, none: +(none)]:
"BNS->IP call address" ↵
GROUP(S) [up to 4 groups separated by commas, none: +(none)]:
umi-rcv ↵
ORIGINATING GROUP NAME SECURITY PATTERN(S)
[comma-separated pattern list, same_as, none: +(none)]: ↵
INITIAL SERVICE STATE [in, out: +(out)]: in ↵
```

Next, **enter sam** is used to establish a SAM504 at a module address actually occupied by the ***UMI***. All 16 boards should be entered if the maximum number of ***UMI*** virtual ports (504) is desired. The following table shows the recommended entries for **enter sam** (port component) for SAM504 ports corresponding to ***UMI originating*** virtual ports:

Field	Value
Protocol	Async
Service Type	Host
Group	<Administered Receive Group>
Cable Type	DCE
Baud Rate	19,200
Parity	Off
Flow Control of SAM by Device	None
Flow Control of Device by SAM	None
AT&T VDM on this Port	No
Permanently Activated Port	No
DCD/DTR leads always high	No
Bits Per Character	8
Number of Stop Bits	1
Endpoint Number or Range	++

```
CC0> enter sam ↵
COMPONENT [module, board, port, ipgate]: mod ↵
MODULE ADDRESS: 2 ↵
COMMENT [up to 60 chars double quoted]:
"Umi" ↵
TYPE [sam8, sam16, sam64, sam504, dt4000]: sam504 ↵
DOWNLOAD SERVER [+(controller)]: ↵
SOFTWARE VERSION [+(standard)]: ↵
TOTAL NUMBER OF BOARDS [1-16]: 16 ↵
TRUNK TYPE [hs, samsl, t1]: t1 ↵
TRUNK SPEED [56k, 64k, 128k, 192k, 256k, 320k, 384k, 448k, 512k, 576k,
640k, 704k, 768k, 832k, 896k, 960k, 1.024M, 1.088M, 1.152M, 1.216M,
1.280M, 1.344M, 1.408M, 1.472M, 1.536M, 1.544M, 2.048M: +(1.544M)]: ↵
SECONDS BEFORE CALL DISCONNECT DUE TO TRUNK FAILURE
[10 second intervals, 20-420: +(50)]: ↵
```

```
CC0> enter sam ↲
COMPONENT [module, board, port, ipgate]: board ↲
MODULE ADDRESS: 2 ↲
BOARD ADDRESS [1-16: +(1-16)]: 1 ↲
BOARD SOFTWARE VERSION [custom, standard: +(standard)]: ↲
INITIAL SERVICE STATE [in, out: +(out)]: in ↲

CC0> enter sam ↲
COMPONENT [module, board, port, ipgate]: port ↲
MODULE ADDRESS: 2 ↲
BOARD ADDRESS [1-16]: 1 ↲
PORT NUMBER [1-32: +(1-32)]: ↲
COMMENT [up to 60 chars double quoted]:
"umi originate vports" ↲
PROTOCOL [async, bisync, ddcmp, hdlc, sdlc, uscope, alc]: async ↲
SERVICE TYPE[console, host, modem, 2way, terminal: +(terminal)]: host ↲
GROUP [up to 8 chars]: umi-rcv ↲
CABLE TYPE [dce, dte: +(dce)]: ↲
BAUD RATE [75, 110, 150, 300, 1200,
          2400, 4800, 9600, 14400, 19200: +(9600)]: 19200 ↲
PARITY [even, odd, off: +(off)]: ↲
FLOW CONTROL OF SAM BY DEVICE [xon/xoff, eia, none: +(none)]: ↲
FLOW CONTROL OF DEVICE BY SAM [xon/xoff, eia, none: +(none)]: ↲
AT&T VDM ON THIS PORT [yes, no: +(no)]: ↲
PERMANENTLY ACTIVATED PORT [yes, no: +(no)]: ↲
DCD/DTR LEADS ALWAYS HIGH [yes, no: +(no)]: ↲
BITS PER CHARACTER [5, 6, 7, 8: +(8)]: ↲
NUMBER OF STOP BITS [1, 2: +(1)]: ↲
INITIAL SERVICE STATE [in, out: +(out)]: in ↲
```

Finally, to enable calls to be set up, **restore sam** is used to put the "SAM504" into service, which puts the ***UMI*** virtual ports into service.

```
CC0> restore sam ↲
COMPONENT [module, board, port, ipgate]: mod ↲
MODULE ADDRESS: 2 ↲
Download in progress for module 2.
Download(s) in progress. <DEL> puts process in background
Download proceeding >
Download complete for module 2.

<UMI> vfy vport all ↲
    Virtual Ports   1 -  32 :
        Type ==> TCP Port w/Call Origination.
        Service State ==> In Service.
        PDD TCP Destination ==> 135.17.59.20  Port 23
```

```
Protocol ==> Asynchronous.
NULL after CR Operation ==> Transparent.
```

A user on the BNS network can now use the dial string “call-ip” to get a connection to the host at IP address 135.17.59.20 and receive telnet service (TCP port 23).

To illustrate an alternative setup, suppose we had not entered the PDD information as part of the **vport** command. The calling user would have gotten the **UMI Destination>** prompt, and could then enter “135.17.59.20 23” to reach the same destination and service. This would be preferable if there were many different endpoints in the IP network which BNS network users need to reach. Rather than require these users to keep track of a multitude of IP addresses, mnemonic names could instead be used. Each of these could be set up with the following command:

```
<UMI> host 1 name=ip-host1 ipaddr=135.17.59.20 port=23 ↵
```

BNS users could now type “ip-host1” at the **UMI Destination>** prompt to reach the same destination/service.

In addition to a **UMI** Predefined Destination within the IP network, and user prompting to reach same as described above, the **UMI** is also capable of **Double Dialing**. This is an advanced feature where the **UMI** Destination is provided as an argument to the BNS dialstring used to reach the **UMI** virtual port.

An example is “nj/bwdev/ddumi.bender” without the double quotes. The first part of the dialstring “nj/bwdev/ddumi” is the BNS address to the **UMI** virtual ports. The virtual ports are defined as BNS_to_IP without a PDD. The name “bender” is the IP destination. Since a mnemonic is used in this example, the name “bender” must exist in the **UMI** host table, or a DNS service must be enabled for resolution. If the name cannot be resolved, the user will be presented with the **UMI Destination>** prompt .

Physical IP address dialing is also supported. The same example could be “nj/bwdev/ddumi.192.168.1.25” which would not require an entry in the host table or DNS resolution. If a specific TCP port is needed, it is appended with a space in the same manner as if the user had entered it manually. eg. “nj/bwdev/ddumi.192.168.7.200 10001”. Please note that when using the *dkcu* command from a UNIX host with a specific TCP port, the entire dialstring must be double quoted for the dkcu command to receive the argument from the UNIX shell correctly. If the BNS dialstring is entered at a BNS Destination prompt, the double quotes should not be used.

The **Double Dialing** feature requires a release level of at least Build 92 of Datakit Release 6 to be installed on the Datakit Node controller. It also requires a release level of at least build 23 to be installed on the **UMI**. The module administration is required to be a **UMI** (as opposed to being administered as a SAM) on the Datakit/BNS node.

A **DTK41** I/O module is required for the use of the **Double Dialing** feature. If a **DTK41** is not present, the user is presented with the **UMI** Destination prompt.

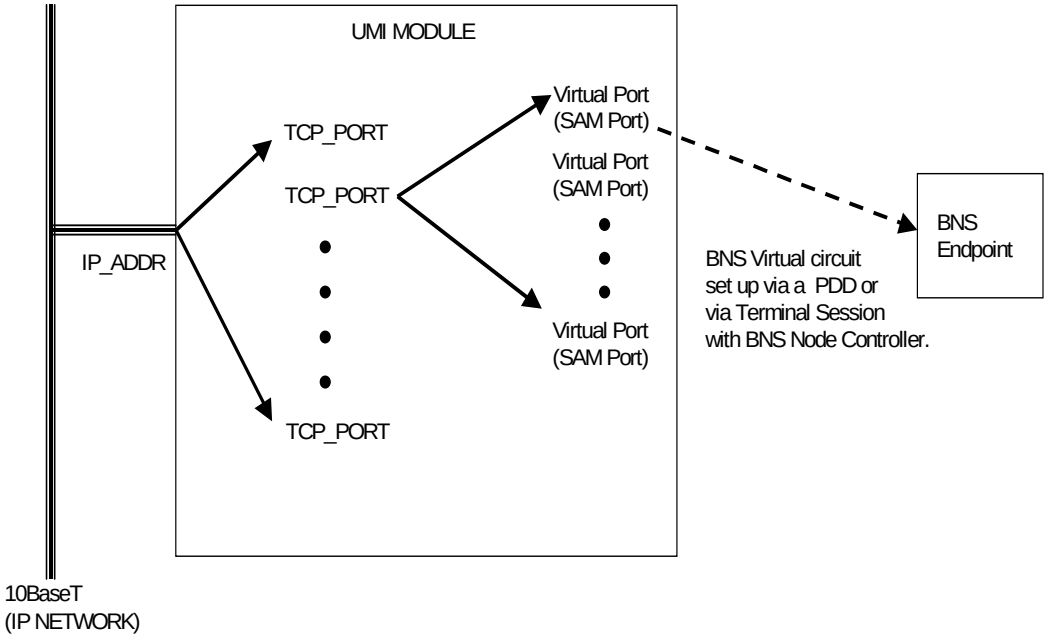
4.3.2 IP-to-BNS Calling

UMI virtual ports designated as ***receive,*** as part of module-based administration, are for the IP-to-BNS calling direction**.** From the node administration perspective, the corresponding SAM ports are set up as ***term*** service-type ports, since they act like terminals originating calls on the BNS

network. If required, PDD information in the form of a BNS destination would be set up as part of node administration for the “SAM504” ports.

A ***UMI*** virtual port configured as ***receive*** “listens” on a configured TCP port for an incoming call arrival. Once a call is established, the Telnet over TCP protocol is used for transport.

The following diagram illustrates the configuration elements affecting call setup in the IP-to-BNS direction.



Referring to the above diagram, the entire BNS network is accessible from anywhere in the IP network via a single IP address. That is the address administered on the ***UMI*** module using the

local command as previously shown (135.17.59.165**)**. Within this address, there are hunt groups of virtual ports, each group having a unique TCP port number (assigned by module administration command **vport**). Virtual ports included in a given hunt group do not need to be contiguous, and the only limit on the group size is the actual number of virtual ports on the ***UMI*** itself (504).

The following command sequence demonstrates the set up of one ***UMI*** hunt group, as well as related node administration for IP-to-BNS calling:

```
<UMI> rm ↵
Remove
MODULE removed from service.
<UMI> vport 33 cnt=32 type=rcv hport=26 ↵
<UMI> rs ↵
Restore
Module restored to service
Reboot Initiated to Make Physical Parameters effective.
```

The first node administration step is to establish an ***originate*** group to which the correct set of SAM504 ports can be associated, as follows:

```
CC0> enter group ↵
GROUP [up to 8 chars]: umi-orig ↵
TYPE [local, trunk: +(local)]: ↵
DIRECTION [originate, receive, 2way]: originate ↵
DEVICE OR HOST [up to 8 chars: +(standard)]: ↵
PASSWORD [up to 8 chars, none: +(none)]: ↵
```

Next, **enter sam** is used to set up the SAM504 ports corresponding to the newly-configured ***UMI*** receive virtual ports. The following table shows the recommended entries for **enter sam** (port component) for SAM504 ports associated with ***UMI receive*** virtual ports:

Field	**Value**
Protocol	Async
Service Type	Terminal
Group	<Administered Originating Group>
Predefined Destination	<PDD for this port if needed>
Cable Type	DCE
Baud Rate	19,200
Parity	Off
Flow Control of SAM by Device	None

Flow Control of Device by SAM	None
Node Echoes User Input	Yes
Call Hold	On
AT&T VDM on this Port	No
Permanently Activated Port	No
DCD/DTR leads always high	No
Connect Time Billing	< As needed>
Attention Character	<Per desired preference>
Attention Action	Command Mode
Bits Per Character	8
Number of Stop Bits	1
Endpoint Number or Range	++

```
CC0> enter sam ↵
COMPONENT [module, board, port, ipgate]: board ↵
MODULE ADDRESS: 2 ↵
BOARD ADDRESS [1-16: +(1-16)]: 2 ↵
BOARD SOFTWARE VERSION [custom, standard: +(standard)]: ↵
INITIAL SERVICE STATE [in, out: +(out)]: in ↵

CC0> enter sam ↵
COMPONENT [module, board, port, ipgate]: port ↵
MODULE ADDRESS: 2 ↵
BOARD ADDRESS [1-16]: 2 ↵
PORT NUMBER [1-32: +(1-32)]: ↵
COMMENT [up to 60 chars double quoted]:
"UMI receive virtual ports" ↵
PROTOCOL [async, bisync, ddcmp, hdlc, sdlc, uscope, alc]: async ↵
SERVICE TYPE [console, host, modem, 2way, terminal: +(terminal)]: ↵
GROUP [up to 8 chars]: umi-orig ↵
PREDEFINED DESTINATION [+(none)]: ↵
CABLE TYPE [dce, dte: +(dce)]: ↵
BAUD RATE [75, 110, 150, 300, 1200,
           2400, 4800, 9600, 14400, 19200, auto: +(auto)]: 19200 ↵
PARITY [even, odd, off: +(off)]: ↵
FLOW CONTROL OF SAM BY DEVICE [xon/xoff, eia, none: +(none)]: ↵
```

```
FLOW CONTROL OF DEVICE BY SAM [xon/xoff, eia, none: +(none)]: ↵
NODE ECHOES USER INPUT [yes, no: +(yes)]: ↵
CALL HOLD [on, off: +(off)]: on ↵
AT&T VDM ON THIS PORT [yes, no: +(no)]: ↵
PERMANENTLY ACTIVATED PORT [yes, no: +(no)]: ↵
DCD/DTR LEADS ALWAYS HIGH [yes, no: +(no)]: ↵
CONNECT-TIME BILLING [on, off: +(off)]: ↵
ATTENTION CHARACTER [none, 1brk, 2brk, del, a character: +(2brk)]: ↵
ATTENTION ACTION [command_mode, disconnect: +(command_mode)]: ↵
BITS PER CHARACTER [5, 6, 7, 8: +(8)]: ↵
NUMBER OF STOP BITS [1, 2: +(1)]: ↵
INITIAL SERVICE STATE [in, out: +(out)]: in ↵
```

After restoring the SAM at module address 2, we observe that the new virtual ports are in service:

```
<UMI> vfy vport all ↵
```

(virtual ports 1 – 32 and 65 – 504 omitted)

```
  Virtual Ports  33 -  64 :
      Type ==> TCP Port 26 w/Call Listen.
      Service State ==> In Service.
      Protocol ==> Asynchronous.
      NULL after CR Operation ==> Transparent.
```

When a TCP/IP connection is made to the ***UMI***'s IP address and TCP port 26, the next available virtual port is selected by round robin. If the SAM port corresponding to the chosen virtual port had been configured with PDD information, the call would automatically progress to the specified endpoint within the BNS network. In this case, the calling user will receive a BNS **Destination>** prompt, and would then enter a valid destination address within the BNS network. See Appendix E for a typical user scenario.

4.4 CLOSED USER GROUPS

CUGs can be established within the BNS network via node administration to control access to BNS endpoints from ***UMI*** virtual ports (represented by their corresponding SAM504 ports), and vice versa *(see Data Networking Products Commands Reference).*

The ***UMI*** also has its own implementation of closed user groups to control access between its virtual ports and endpoints on the IP network. The module administration command **cug** is used to create a closed user group as a single IP address or range of addresses in a sub net. The **vport** command allows up to 128 CUGs to be associated with a group of virtual ports. The **console** command allows up to 128 CUGs to be associated with the telnet administrative console of the UMI. The **snmp** command allows up to 128 CUGs to be associated with the SNMP interface to the UMI as a security feature. Calls in either the IP->BNS direction are restricted as follows:

4.4.1 IP-to-BNS Calling

A call to the TCP port number corresponding to a hunt group of ***rcv***-type virtual ports will be blocked unless the calling IP address belongs to at least one of the CUGs associated with the selected virtual port. End-to-end security could be accomplished with node administration, by

setting up a PDD to a specific BNS destination address for the SAM504 ports corresponding to the virtual port hunt group (most restrictive), or by using CUG assignments in the BNS network to control dialing access from each originating group of SAM ports. See Appendix F for an example scenario.

4.4.2 Telnet Console

A call to the TCP port number corresponding to the telnet console (i.e. 1023) will be blocked unless the calling IP address belongs to at least one of the CUGs associated with the telnet console.

4.4.3 SNMP Interface

A UDP packet sent to the SNMP interface port number will be blocked unless the sender IP address belongs to at least one of the CUGs associated with the SNMP interface.

4.5 UMI CONFIGURATOR

This tool facilitates ***initial UMI*** configuration. The configurator is a free StarKeeper® II NMS utility that significantly reduces the time required to configure an entire ***UMI***, including node and module-based configuration. ***UMI Configurator*** Installation and Usage notes can be found in Appendix D.

5 UMI MODULE COMMAND REFERENCE

These module commands are used to configure the operation of the ***UMI***.

Not all commands are visible all the time. Should the unit be logged out, only the **login** command is visible. The **reboot** command places the unit in the logged-out mode.

Parameters of the form **name=<value>** must be entered in order, and all parameters being used must be entered on a single command line.

- Commands may be entered in upper or lower case.
- Parameters of the form **name=<value>** may use upper or lower case for **name**.
- Case is preserved for values.
- Backspace erases one character.

5.1 LOGIN

Syntax: login passwd=<password> `(default password is: initial)`

The **login** command is only visible when the unit is in the logged out *(i.e. secure)* mode. The unit enters this mode whenever a logout command is issued or when a reboot occurs for any reason. The password is **not echo-suppressed** and consists of up to seven alphanumeric characters. Special characters are not allowed.

NOTE: The "Login" command accepts an optional form. By entering the command without arguments, a password will be prompted for. In this alternate form, the password is not echoed. Only asterisks are echoed (i.e. `login <enter>`***).***

5.2 LOGOUT

Syntax: logout

The **logout** command is only visible when the unit is logged in. The command has no arguments. It returns the user to the logged out mode.

5.3 CHGPASS

Syntax: chgpass old=<password> new=<password> confirm=<password>

The **chgpass** command is only visible when the unit is logged in. The command allows the user to change the password configured for the unit. All arguments are required for the command to complete. The old password is the one currently in effect. The new and confirm passwords must be identical. Passwords are up to seven characters in length. The characters are alphanumeric and special characters are not allowed.

5.4 LOCAL

Syntax: local [ipaddr=<IP address>]
[submask=<submask>]
[tcpunreach=< ICMP | RESET >]

The **local** command is only visible when the unit is logged in.

The **ipaddr** field is the IP address of this unit.

The **submask** field is the subnet mask for the LAN segment on which the unit is located. It defaults to 255.255.255.0.

The IP address and subnet mask are used to determine whether a destination IP address is on the same LAN segment, or if a gateway hop is required

The operation of the **UMI,** when it is called to an invalid TCP port, is specified with the **tcpreach=<ICMP | RESET>]** parameter. When set to **ICMP**, the caller is sent an "ICMP Port Unreachable" message. When set to **RESET**, the TCP connection is sent a TCP reset to the initiator.

5.5 GATEWAY

Syntax: gateway ipaddr=<IP address>

The **gateway** command is only visible when the unit is logged in.

The **ipaddr** field is the IP address of the gateway router to be used to reach a destination IP address on a different LAN segment.

5.6 DOMAIN NAME SERVER

Syntax: dns [ipaddr1=<IP address>]
[ipaddr2=<IP address>]
[ipaddr3=<IP address>]
[name1=<Domain Name>]
[name2=<Domain Name>]
[name3=<Domain Name>]

The **dns** command is only visible when the unit is logged in.

Each **ipaddrX** field is the IP address of a Domain Name Server to be used for mnemonic addresses not defined in the host table. When all are set to 0.0.0.0, the DNS functions on the ***UMI*** are disabled. The DNS addresses are used in order. If only one address is to be defined, it should be **ipaddr1**.

The **name1**, **name2**, and **name3** parameters are domain names. These domain names are appended to a dial string which is not fully specified for DNS purposes. For example, a name "bender.ho.lucent.com" is fully specified, so nothing is appended by the ***UMI***. A name such as "bender" would need to have a domain appended before the DNS server could resolve it. The ***UMI*** will append the specified domain names in the order of **name1** through **name3**, and send the resulting strings to the DNS server in succession until the latter is able to perform a resolution.

5.7 TACACS+ RADIUS Servers

Syntax: tac < PRI | SEC > [ipaddr=<IP Address>]
[port=<TCP Port>]
[key="Encryption Key" | NONE]
[ENABLE]
[DISABLE]

The **tacplus** command is only visible when the unit is logged in. The tac command allows the configuration of up to two **TACACS+** RADIUS servers for the device. the servers are used as a primary server and a secondary server, although they may be individually disabled.

The **< PRI | SEC >** syntax specifies which server is to be configured. A server may not be configured while enabled

The **[ipaddr=<IP Address>]** specifies the IP address of the configured server.

The **[port=<TCP Port>]** specifies the TCP port to use when communicating with the server. The TACACS+ service defaults to TCP port 49, but any port may be specified.

The **[key="Encryption Key" | NONE]** specifies an encryption key to use. The Encryption key must be enclosed in double quotes, and the double quotes are not part of the key. If no encryption is desired, the value of **NONE** is used to designate unencrypted service.

The **ENABLE** command allows this server to be used for service, and prevents further configuration.

The **DISABLE** command prevents this server from being used for service, and subsequently allows configuration.

5.8 HELP

Syntax: help

The **help** command is always visible. It displays the commands allowed for the mode that the unit is currently in.

5.9 VERSION

Syntax: version

The **version** command is only visible when the unit is logged in. It displays the current software and database revisions of the unit. If the UMI has been upgraded, and not yet rebooted; the version command will also display the version number of the software staged for operation.

5.10 REBOOT

Syntax: reboot

The **reboot** command is only visible when the unit is logged in. It resets the unit, which allows physical attributes to be set. The console interface returns to the logged-out mode.

This command is password protected. The administrative password will be prompted and echoed with asterisks.

5.11 REMOVE MODULE

Syntax: rm

The **remove** command is only visible when the unit is logged in. It takes the unit out of service. ***This command must be performed before any module level configuration changes can occur.***

This command is password protected. The administrative password will be prompted and echoed with asterisks.

5.12 RESTORE MODULE

Syntax: rs

The **restore** command is only visible when the unit is logged in. It returns the unit to service. If any physical attribute was changed, the unit will be automatically rebooted by this command.

5.13 CLEAR MEASUREMENTS

Syntax: `clr`

The **clear** command is only visible when the unit is logged in. It sets all the measurements and error counters to zero.

5.14 DISPLAY MEASUREMENTS

Syntax: dm [mod][vport<vport#>]

The **dm** command is only visible when the unit is logged in. It displays the current measurements in a formatted report on the console.

The value of **<vport #>** is between 1 and 504, inclusive.

See Appendix A for an enumeration of module measurements. Virtual port information is not displayed on a module-level report.

See Appendix B for an enumeration of virtual port measurements. Module information is not displayed on a virtual-port-level report.

5.15 VERIFY

Syntax: vfy [mod][vport<vport#|vport range|ALL>][cug][host]

The **vfy** command is only visible when the unit is logged in. The command accepts **mod, vport, cug, or host** as objects.

The **mod** option displays the module-level configuration in a formatted report.

For the **vport** option, the **<vport #>** is between 1 and 504. The **<vport range>** accepts a numeric range such as 3-6.

A value of **ALL** provides a report of all virtual ports in the ***UMI***.

The **cug** option produces a list of all closed user groups on the ***UMI***, with the specification of how each is configured.

The **host** option produces a list of all mnemonic host names configured on the ***UMI***.

The **vfy** command with a large range, or the value of "**ALL**", may produce voluminous results. The output of the **vfy** command may be manually aborted with a **Control-C**, or a **DEL**, character.

5.16 HOST NAME ADMINISTRATION

Syntax: host <host #> [name=<host name>]

[ipaddr=<IP address>]
[port=<TCP port>]
[del]

The ***UMI*** can maintain a set of mnemonic host names for "dialing" out to the IP network, analogous to the /etc/hosts file on both UNIX and Microsoft Windows platforms. This allows the ***UMI*** to perform a translation between a user-provided name and its associated IP address and TCP port number during call setup, for originating virtual ports without a PDD. The use of a mnemonic name is optional; the ***UMI*** will always accept an IP address in its base form.

The **host** command is used to configure the translation table.

The **name** field is a mnemonic up to 9 characters in length.

If the parameter **del** is used, the entry is deleted.

5.17 Mnemonic PDD ADMINISTRATION

Syntax: pdd <pdd #> < <Mnemonic PDD Address> | DELETE >

The ***UMI*** can maintain a set of mnemonic destination addresses for "dialing" out to the IP network. These can be assigned to individual virtual ports via the ***vport*** command. The mnemonic address may be an entry in the host table, an unqualified DNS address (if domain names have been defined), or a fully qualified DNS address. The mnemonic address may be up to 63 characters in length.

The use of mnemonic addressing is optional in the UMI. Numeric IP addresses may be defined on a virtual port range basis using the ***dest*** parameter of the ***vport*** command.

The **pdd** command is used to configure the translation table. Up to sixteen mnemonic destinations may be defined. The mnemonic destinations may be assigned to any number of virtual ports.

The **<pdd_id>** parameter is numeric identification of the mnemonic address. It has a value of one through sixteen inclusive.

The **<Mnemonic PDD Address>** is the address string without any quotes.

If the **<Mnemonic PDD Address>** parameter of **delete** is used, the entry **<pdd #>** is deleted.

5.18 SNMP

Syntax: snmp [ipaddr=<SNMP Mgr IP address>]
[port=<SNMP manager trap port>]
[CUG=<<+|-> CUG Number>]
[PUBLIC=< YES | NO >]
[COMM=< "Double Quoted Community String" | NONE >]
[syscontact=< "Double Quoted String" | NONE >]
[sysname=< "Double Quoted String" | NONE >]
[sysloc=< "Double Quoted String" | NONE >]

The **snmp** command is used to configure the IP address of the SNMP trap manager. Since traps are unsolicited alarms, an agent can take the initiative to inform the manager of the occurrence of a predefined condition.

A single or multiple SNMP managers can access the ***UMI***. However, only one SNMP manager can be predefined as the trap manager. As result of using this command, all traps will be directed to the chosen trap manager. The **ipaddr** field defines the IP address of the SNMP manager to whom the traps are to be sent. The **port** field indicates the UDP port on that SNMP manager and defaults to the standard value of 162. If closed user groups are to be defined on the interface the **cug** option allows the association of up to 32 individual groups defined with the **cug** command.

The **COMM** option allows the configuration of a private SNMP community. The *public* community is always present per the specification. This option takes effect immediately.

The **UMI** allows setting of an SNMP community in addition to the *public* community. The *public* community is recognized when the **[PUBLIC=YES]** option is selected. Recognition of the public community is the default operation. When **[PUBLIC=NO]** is selected, the *public* community is not recognized.

The **sysContact** option allows a non-volatile initialization of the MIB-II sysContact variable. The MIB-II sysContact variable remains changeable from the SNMP management station. However, it is initialized to the non-volatile value each time the UMI is rebooted. Setting this option to the constant *none* removes the non-volatile initialization of the MIB-II variable. This option takes effect immediately. However, the non-volatile contents are stored on the DTK41. If a DTK41 is not installed, the settings are lost on the next reboot. If the UMI is administered via the Datakit Controller enhanced interface updated for build #21 or later, the DTK41 is not required since the Datakit controller will apply the settings on each restart.

The **sysName** option allows a non-volatile initialization of the MIB-II sysName variable. The MIB-II sysName variable remains changeable from the SNMP management station. However, it is initialized to the non-volatile value each time the UMI is rebooted. Setting this option to the constant *none* removes the non-volatile initialization of the MIB-II variable. This option takes effect immediately. However, the non-volatile contents are stored on the DTK41. If a DTK41 is not installed, the settings are lost on the next reboot. If the UMI is administered via the Datakit Controller enhanced interface updated for build #21 or later, the DTK41 is not required since the Datakit controller will apply the settings on each restart.

The **sysLocation** option allows a non-volatile initialization of the MIB-II sysLocation variable. The MIB-II sysLocation variable remains changeable from the SNMP management station. However, it is initialized to the non-volatile value each time the UMI is rebooted. Setting this option to the constant *none* removes the non-volatile initialization of the MIB-II variable. This option takes effect immediately. However, the non-volatile contents are stored on the DTK41. If a DTK41 is not installed, the settings are lost on the next reboot. If the UMI is administered via the Datakit Controller enhanced interface updated for build #21 or later, the DTK41 is not required since the Datakit controller will apply the settings on each restart.

5.19 INSTALL (Software Registration)

Syntax: install [key=<software key>]
[fpkey=<software key>]

The **UMI** has a unique device software key, and multiple per port feature package keys. This section is included in the user manual for completeness. Under normal circumstances, only the mechanized utilities utilize this command. It may be executed manually under an emergency situation. Depending on the device, the keys may or may not be installed by the factory. The per port feature package keys my be added at any time, and do not affect the operation of the unit.

The registration procedure does not require a restart to take effect on a device running the registered software.

When executed without arguments, the **install** command will display the significant information needed to manufacture the device software key. The device IP address may also be required. No additional information is needed to create the feature package keys.

The **key=<software key>** argument allows the entry of an eight-character alphanumeric software registration that is unique to this **UMI** device. If an invalid key is entered, a MINOR alarm is generated to that effect. The passwords are not altered. The **rstpass** command has been created to reset the passwords should that become necessary.

The **fpkey=<software key>** argument allows the entry of an eight-character alphanumeric software registration that is unique to a port, and software feature package, on this **UMI** device for the current software build. The specific feature package referenced by the software key becomes immediately available on the port without subsequent download. The <software key> has effect on only one port. Other ports on the device are not affected. If the same software feature package is needed on multiple ports, then multiple feature package keys are applied.

The **install** command is always available.

5.20 RSTPASS (Resetting the Password)

Syntax: rstpass [key=<Password Key>]

The **rstpass** command is a command whose function is to reset the password(s) of the device to factory default values. This function was formerly performed as part of the software registration. Breaking it out into a separate command allows the software to be registered without password updates to take place.

When invoked without arguments, the **rstpass** command will display the relevant information needed to generate the **<Password Key>**. This information is relayed to the technical support staff. The generated key is then used with the **key=<Password Key>** argument. The **rstpass** command should not be run between the time the key data is generated and the **<Password Key>** is utilized. Similarly, if the device is restarted, the resultant **<Password Key>** will not perform its intended function.

5.21 CONSOLE TIMEOUT

Syntax: timeout [off | <number of minutes>]

The ***UMI*** console uses a three-wire interface (RD, TD, GND), and the lead state of other signals is not relevant. This would imply that the only way to change the state of the console is to explicitly log in or log out, or via a reboot or reset, which forces the console to be logged out.

For users who wish the console to automatically log off after a period of inactivity, there is a console timer. The console timer defaults to the disabled condition, and may be activated by the **timeout** command.

This command is only visible when the console is logged in. The **<number of minutes>** value must be between 1 and 255, inclusive.

When the ***UMI*** determines a period of inactivity of the specified time, it automatically forces the console to log off. An **INFO**-level alarm (see Appendix C) is issued at that time.

5.22 VIRTUAL PORT ADMINISTRATION

Syntax: vport <vport #> [cnt=<#>]
[incr=<#>]
[type=<ORIG|RCV>]
[pdd=<PDD #>]
[dest=<IP Addr>]
[dport=<Dest TCP Port>]
[hport=<Hunt Group TCP Port>]
[cug=[+|-]<CUG Number>]
[prot=<ASYNC | SYNC | RAW | VPAD>]
[crfix=< TRANS | NONULL >]
[crlf-< TRANS | LCS60 >]
[data=<7bit | trans>]
[sess=<hold | trans>]
[crypt=<on | off>]

The **vport** command configures one or more virtual ports. The value of **<vport #>** is in the range of 1 through 504, inclusive, and represents the first virtual port to be affected by this command.

The **cnt** parameter allows more than one port to be affected. All virtual ports may be configured with a single command having a **cnt** of 504.

The **incr** parameter allows for multiple **dport** configurations from a single command. It is an increment added to the TCP port base value for each subsequent port specified in the **cnt** option.

Examples using the inrc command together with the cnt command.

The first example configures two vports (starting at vport 1) and increments the dport by a value of one. The second example configures three vports (starting at vport 1) and increments each subsequent dport by a value of 200.

Note: The "incr" option is used to configure the "dport" option only.

Example 1

```
<UMI> vport 1 cnt=2 incr=1 dest=192.168.59.251 dport=20000

<UMI> vfy vport 1
    Virtual Ports   1 -   1 :
```

```
        Type ==> TCP Port w/Call Origination.
        Service State ==> In Service.
        PDD TCP Destination ==> 192.168.59.251  Port 20000
        Protocol ==> Asynchronous.
        NULL after CR Operation ==> Transparent.
        LF after CR Operation ==> Transparent.

<UMI> vfy vport 2
    Virtual Ports   2 -   2 :
        Type ==> TCP Port w/Call Origination.
        Service State ==> In Service.
        PDD TCP Destination ==> 192.168.59.251  Port 20001
        Protocol ==> Asynchronous.
        NULL after CR Operation ==> Transparent.
        LF after CR Operation ==> Transparent.
```

Example 2

```
<UMI> vport 1 cnt=3 incr=200 dest=192.168.17.7 dport=30000

<UMI> vfy vport 1
    Virtual Ports   1 -   1 :
        Type ==> TCP Port w/Call Origination.
        Service State ==> In Service.
        PDD TCP Destination ==> 192.168.17.7  Port 30000
        Protocol ==> Asynchronous.
        NULL after CR Operation ==> Transparent.
        LF after CR Operation ==> Transparent.

<UMI> vfy vport 2
    Virtual Ports   2 -   2 :
        Type ==> TCP Port w/Call Origination.
        Service State ==> In Service.
        PDD TCP Destination ==> 192.168.17.7  Port 30200
        Protocol ==> Asynchronous.
        NULL after CR Operation ==> Transparent.
        LF after CR Operation ==> Transparent.

<UMI> vfy vport 3
    Virtual Ports   3 -   3 :
        Type ==> TCP Port w/Call Origination.
        Service State ==> In Service.
        PDD TCP Destination ==> 192.168.17.7  Port 30400
        Protocol ==> Asynchronous.
        NULL after CR Operation ==> Transparent.
        LF after CR Operation ==> Transparent.
```

A virtual port which waits for an incoming call from the IP network is designated by **type=rcv**. For a virtual port which originates calls to the IP network (**type=orig**), the PDD of the call is defined by either the **pdd=<pdd#>** or by **dest=<ipaddr>** and **dport=<tcp_port>**. These destination options are ***only required for PDDs***. The two forms are mutually exclusive. BNS users setting up

calls to originating virtual ports without a PDD specification will be presented a user interface for "dialing" a destination IP address and TCP port.

When a virtual port is a call receiver *(listener)*, it is assigned a default port number value of 23 *(TELNET service)*. If only one class of service is needed for access to the BNS network, this does not need to be changed. However, when a specific TCP port is specified via the **hport=<tcp_port>** option, it is used in place of the default value.

Multiple virtual ports may share the same TCP port value, to define a **hunt group** of virtual ports. A connection that is directed to this TCP port value would select the next available virtual port. The **hport=<tcp_port>** option may only be used with receive virtual ports.

The **cug=[+|-]<CUG_num>** option allows the inclusion or deletion of a Closed User Group in the list of CUGs assigned to the virtual port. The "**+**" will add the **<CUG_num>** to the CUG list. The "**-**" is used to delete the **<CUG_num>** from the list.

The **prot** option allows the specification of the encapsulation method associated with the virtual port. The **async** encapsulation method uses a telnet service without extensions. This is applicable to asynchronous connections. The **sync** encapsulation is basically the same as asynchronous, but with the extensions needed for synchronous protocols. The **raw** option is used when no encapsulation is desired. It should be noted that the DT-4000 and DT-2020 series devices are tolerant of synchronous encapsulation for asynchronous connections, so virtual ports with these protocols may be grouped together for connections exclusively to endpoints on those devices. The **vpad** option selects a virtual pad service. This allows IP resident entities to access devices on the DK/BNS network with a mediation service similar to the DT-xx8x devices.

The **crfix=< TRANS | NONULL >** option accommodates an anomaly in some early variants of telnet implementation on UNIX systems, which insert a NULL character in the data stream after a carriage return. Most end devices are not affected by this NULL character. However, some devices (e.g. the BNS control computer) have erroneous operation if these characters are received. The value **TRANS** indicates transparent operation, where all data received by the ***UMI***, including a NULL after a carriage return, is forwarded to the end device. The value of **NONULL** removes a NULL character immediately following a carriage return. No other NULL characters are affected. The default operation is transparent, and the **crfix** option may only be specified if the protocol selected is asynchronous.

The **crlf=< TRANS | LCS60 >** option accommodates Microsoft MSDOS (and Windows variants) of telnet implementations. These implementations insert a LF character in the data stream after a carriage return. Since both characters are treated equally by some endpoints, the result is a double line entry where only one was desired. The LCS60 device would always strip the LF following a CR. However, this would yield problems for some applications where transparency was desired. The **crlf** option allows the selection of either operation. When the **crlf=TRANS** is selected, the virtual port is transparent. When the **crlf=LCS60** is selected, the virtual port performs LCS60 style processing on the LF following a CR. It is stripped from the data stream.

The **data=<7bit | trans>** option allows you to "filter" data to 7bit by essentially masking out the parity bit on each character. This is useful when a TY card connected to a Network Element is configured as 8 bit and no parity and the Network Element is 7bit and even parity. Datakit takes care of parity conversion on egress, but Telnet does not. Some telnet clients will display garbage, and others will work fine. This feature will eliminate the issue altogether.

The **sess=<hold|trans>** option allows an IP session to be maintained across multiple BNS sessions. The option is available for sessions that originate in the IP infrastructure, and for the *async* encapsulation method. The default value is **trans** which provides seamless service instead of session hold. If the BNS port has been configured with an *attention character*, that *attention*

character will control the BNS sessions. The UMI supports *attention character* sequences that are printable text, a DELETE, or a single Break. The double break attention sequence is not supported.

The **crypt=<on|off>** option allows an IP session to be encrypted between peer entities. Both ends of the connection must have encryption enabled or disabled. The UMI uses a dynamic random key method of data encryption when this option is enabled.

5.23 VIRTUAL PAD ADMINISTRATION

Syntax: vpad <vport#> [cnt=<#>] [padecho=<ON | OFF>]
[paderase=< NONE | BS | <HEX BYTE>]
[padidle=<#X.3 Ticks>]
[padparity=<TRANS | EVEN | ODD>]
[padcrlf=<NONE | RMT | VC | BOTH>]
[padfwd=<NONE | CR | CRDROP |
SEMI | EXCL | ALL | GRPx >]
[padcmap=<ON | OFF>]
[padapi=<TELNET | RAW>]
[paddial=<DEL | <DK/BNS Dest> >]

This command configures the virtual pad options of a vport. The options are only relevant when the vport has been set to **prot=vpad** on the vport command.

The **[cnt=<#>]** parameter allows more than one port to be affected. All virtual ports may be configured with a single command having a **cnt** of 504.

The **padecho=< ON | OFF >** refers to reference #2 in the X.3 parameter list. When set to **OFF**, the PAD will not echo characters back to the IP endpoint. When set to the value of **ON**, all characters are to be echoed back to the IP source.

The **paderase=< NONE | BS | <HEX BYTE> >** option specifies reference #16 in the X.3 parameter list. It is used with manual telnet connections to an X.25 VC. It sets the buffer editing “erase” character. When the special “erase” character is received by the PAD for a specific virtual circuit, the previous character in the packet accumulation buffer is deleted. If the **padecho** option was also enabled, a “Backspace Blank Backspace” sequence is emitted to the user. When the **paderase** option is set to NONE, the PAD will not have a special “erase” character. When the value is BS, it is set to the ASCII backspace character 0x08. Otherwise, any character may be entered as a hexadecimal byte in 0xXX notation.

The **padfwd=<NONE | CR | CRDROP | SEMI | ALL | GRPx>** option specifies reference #3 of the X.3 parameter list. This is the forwarding condition (outside the PAD timer) which will forward data towards the X.25 virtual circuit. A value of **NONE** indicates that there are no character forwarding conditions. A value of **CR** indicates that a carriage return will forward any accumulated data (including the carriage return). A value of **CRDROP** indicates that a carriage return will forward any accumulated data (but not including the carriage return). A value of **SEMI** indicates that a

semicolon will forward any accumulated data including the semicolon. A value of **EXCL** indicates that an exclamation point will forward any accumulated data including the exclamation point. A value of **ALL** indicates that all data is to be forwarded immediately. The **ALL** option has the effect of generating single user character X.25 packets on this virtual circuit. The **GRPx** values specify selected groups of forwarding characters. **GRP1** forwards on ESC, BEL, ENQ, and NAK. **GRP2** forwards on DEL, CAN, DC2. **GRP3** forwards on ETX, EOT. **GRP4** forwards on HT, LF, VT, and FF. Multiple forwarding conditions are allowed simultaneously. Setting **padfwd** to a value aggregates with the previous value of **padfwd**. The **padfwd=none** is required to clear the forwarding conditions.

The **padidle=<#X.3 ticks>** parameter refers to reference #4 of the X.3 parameter list. This is the time forwarding condition. When it expires, it will forward any data collected to the X.25 circuit. The timer is reset to the specified timer value whenever a forwarding condition is reached. The value is based on ticks of 1/20th of a second each per the X.3 specification.

The **padparity=< TRANS | EVEN | ODD >** parameter is not present in the X.3 parameter list. It allows special parity treatment for interface to network elements that require parity. The default value is transparent operation. The value of **TRANS** sets the operation to be transparent. When the parity treatment is transparent, the data is not modified in either direction. The value of **EVEN** sets the operation to be even parity towards the (B)X.25 device, and stripped parity towards the TELNET. The value of **ODD** sets the operation to be odd parity towards the (B)X.25 device, and stripped parity towards the TELNET.

The **padcrlf=<NONE | RMT | VC | BOTH>** parameter refers to reference #13 of the X.3 parameter list. This is the action to be taken when a CR is received in the data stream from the remote IP endpoint. A value of **NONE** indicates that there is to be no LF (line feed) insertion. A value of **RMT** will insert an LF following a CR whenever it is sent towards the remote IP endpoint. A value of **VC** will insert an LF following a CR whenever it is sent towards the X.25 virtual circuit. A value of **BOTH** will insert an LF following a CR in either direction.

The **padcmap=< ON | OFF >** option provides the automatic case mapping from lower case to upper case. When the option is set to **ON**, all lower case characters are automatically converted to upper case. When **OFF**, no transformations are performed.

The **padapi=< TELNET | RAW >** option provides a means of selecting the PAD virtual circuit to use **raw** protocol. The **raw** protocol is essentially asynchronous, but without the benefit of Telnet RFC encapsulation. It is used for applications that do not implement the Telnet RFC. The default for this option is to use the Telnet encapsulation.

The **paddial=< DEL | <DK/BNS Destination String>** option provides a means of having the UMI VPAD dynamically dial the datakit node. This is useful for module sparing purposes as the configuration is stored on the module and may be individually moved using the backup and restore utilities. When dynamic dialing is used, the DKCC "sam port" that corresponds to the UMI VPORT used by the VPAD should be set to Asyncrhonous, Terminal, Any Baud Rate except not autobaud, Not PAP, No PDD present, and the node should not echo the user input. It should be noted that all the DKCC ports may be configured identically. All 32 ports on a SAM "board" may be configured with one command sequence.

5.24 VIRTUAL PAD SNOOPER

Syntax: snoop OFF
snoop VP <virtual port #> [verbose]

This command allows the display of the data flow for a virtual port configured as a virtual pad. The command will issue an error if the virtual port is not configured as a virtual pad.

The **snoop OFF** form will disable any snooping this is presently enabled on all virtual ports.

The **snoop VP <virtual port#> [verbose]** form will enable snooping on the specified virtual port number. If verbose is specified, then the actual data contents are also displayed. Otherwise, the frame (grouping) indication and size is displayed. Repeatedly invoking this command enables snooping on multiple virtual ports. However, the output of the snooper can be exceptionally large even when the verbose option is not used.

5.25 CLOSED USER GROUP (CUG) ADMINISTRATION

Syntax: cug <CUG_num>[ipaddr=<IP address>][submask=<IP submask>]

The **cug** command is only visible when the unit is logged in. The **<CUG_num>** parameter is the closed user group identifier used to assign the CUG to a virtual port (with the **vport** command), and may be a value between 1 and 32, inclusive. The CUG may also be assigned to the telnet console (with the **console** command), or the SNMP interface (with the **SNMP** command).

Each CUG is specified by a single IP address and subnet mask pair. The **ipaddr** parameter is an address of an endpoint (or base address of a group of endpoints) to be allowed into the group. The **ipaddr** value *ANDed* with the **submask** value must agree with the caller's or destination's IP address *ANDed* with the same **submask** for a call to be allowed to or from a virtual port to which the CUG is assigned. Depending on the **submask** value, this allows an individual (submask=255.255.255.255), intermediate, or network-wide level of authorization.

Setting the **ipaddr** value to 0.0.0.0 deletes any prior configuration for the **<CUG_num>**. A **<CUG_num>** may not be deleted if it is currently assigned to any virtual port.

A list of all configured CUGs is reported via the **vfy cug** command. The list of closed user groups associated with a given virtual port is presented in response to the **vfy vport** command.

5.26 DISPLAY CONNECTIONS

Syntax: dconn

The **dconn** command is only visible when the unit is logged in. The command displays the connections between virtual ports and their destinations. Only active virtual ports are displayed.

5.27 DISCONNECT PORT

disc VPORT [<port_num>] [CONSOLE]

The **disc** command is only visible when the unit is logged in. The value of **<port_num>** is between 1 and 504, inclusive. If the specified virtual port is in service, any existing circuit spanning it will be dropped. This is useful in IP networks when the remote peer vanishes due to a remote reboot or a network error.

The value of **console** will allow the disconnect of the telnet console from itself or the serial console interface. This is useful when the telnet console peer is not responding.

This command is password protected. The administrative password will be prompted and echoed with asterisks.

5.28 PING

ping <IP address> [Interval Seconds]

The **ping** command is only visible when the unit is logged in. The command has a single argument, the IP address that is to be pinged.

The **ping** command formats an ICMP echo request packet which is then sent to the IP Address specified. The device with that address will issue an ICMP echo reply to the request. This is required of all IP implementations by RFC 791. If a reply is received, an informational alarm is issued on the UMI console. If no reply is received, there is a timeout message that will appear for that ICMP echo request.

The ping command issues a single ICMP echo request packet and awaits a response. The response is printed, and another ICMP echo request is issued. The operation continues until the user presses *any* character. The **[Interval Seconds]** argument specifies the amount of time to wait in seconds between the individual ICMP echo requests.

It should be noted that some host Internet Protocol implementations issue duplicate responses to a single ICMP request. The **ping** command will suppress duplicate replies.

5.29 TraceRoute

trte <IP address>

The **trte** command is only visible when the unit is logged in. The command has a single required argument, the IP address that is to be pinged.

The **trte** command formats an ICMP echo request packet that is then sent to the IP Address specified. The valid packet "time to live" is set to an initial value of "1". If the IP address is on the local subnet, the ICMP echo will respond immediately. If the IP address is on a different subnet, the gateway router will decrement the "time to live" upon routing the packet. When the "time to live" reaches zero, the gateway sends an ICMP "time exceeded" message to the **xxxx**. The **xxxx** then displays the gateway device, and increments the "time to live" on the next ICMP echo request packet. This continues until the IP address is reached. The result is a display of all the intermediary gateway devices used to reach the IP address from the **xxxx**.

If no answer is received, each "time to live" value is tried 3 times before an increment. The timeout is 5 seconds for each attempt. The maximum number of "time to live" is set to 30 in this build of the **xxxx**.

Since a traceroute command can be unusually long in duration, any character sent to the console will interrupt the operation of the traceroute command.

5.30 Label

Syntax: label ["Double Quoted String" | none]

The **label** command is used to give the command console a unique prompt. The command is visible only when logged into the UMI administrative console. If the **label** command is invoked without arguments, the current configuration of the label is displayed. If the argument to the **label** command is the word 'none', any current label is set to a null value. If the argument to the **label** command is a double quoted string, the contents of the string becomes the application console

prompt label. A console label may be up to thirty one characters in length, and may contain spaces. The console label string may not contain the colon character (:).

5.31 Administrative Password

Syntax: admpass [old=<Old Password>]
new=<New Password> confirm=<New Password>

The **admpass** command is used to configure or change an administrative password for the UMI console. The administrative password is required to use the ***Remove***, ***Disconnect***, or ***Reboot*** commands. The administrative password defaults to a NULL value until changed.

The **[old=<Old Password>]** specifies the previous administrative password for use of this command. This option is not required on the first configuration of the administrative password.

The **new=<New Password> confirm=<New Password>** specifies the administrative password. Both values are required and must match exactly.

The administrative password takes effect immediately after the successful execution of the command.

5.32 CONSOLE Administration

Syntax: console [cug=<<+|->CUG Number>]

The **console** command is used to configure or change the closed user group configuration of the telnet console to the UMI. Up to 32 CUGs may be associated with the telnet console.

5.33 HPIO Administration

Syntax: HPIO [RESET]
[ENABLE < ALL | FIBER | <10/100 PHY RANGE> >]
[DISABLE < ALL | FIBER | <10/100 PHY RANGE> >]
[AUTO < ALL | <10/100 PHY RANGE>]
[10HDX < ALL | <10/100 PHY RANGE>]
[10FDX < ALL | <10/100 PHY RANGE>]
[100HDX < ALL | <10/100 PHY RANGE>]
[100FDX < ALL | <10/100 PHY RANGE>]

The **HPIO** command is used to enable or disable the physical LAN connections on the DTK41 high performance I/O module. The DTK41 operates defaults to all ports being active. Therefore, no configuration is required unless security of the unused ports becomes an issue.

The **RESET** command option is used to perform a restart on the DTK41 module. Under normal circumstances, it should never be required to use this command. No arguments are required. The **'Link Active'** alarms for all connected links will be issued after a reset.

The **ENABLE** command enables the physical interfaces that have been previously disabled. The DTK41 defaults to all interfaces being enabled. The **ENABLE** accepts a target to specify those ports to be affected. A target value of **ALL** specifies that all three 10/100 BaseT ports, and the Fiber port is to be affected. A target value of **FIBER** specifies that only the fiber port is to be affected. A numeric value or range specifies the 10/100 BaseT ports. The numeric value may be a single number (e.g. '2') or a range (e.g. 1-2).

The **DISABLE** command disables physical interfaces. When disabled, the interface is not capable of communications. The **DISABLE** accepts a target to specify those ports to be affected. These are in the same form as the ENABLE command.

The DTK41 10/100 PHY ports default to automatic negotiation. This can also be set with the **AUTO** command option. The 10/100 ports can be set to a *static* configuration. When set to a *static* configuration, the automatic negotiation advertisement of the DTK41 10/100 PHY is limited to the selected option. The four *static* configurations are: **10HDX** for 10Mbps half duplex, **10FDX** for 10Mbps full duplex, **100HDX** for 100Mbps half duplex, and **100FDX** for 100Mbps full duplex.

It is **strongly recommended** that the default configuration of automatic negotiation (**AUTO**) is used. Any forced mismatch between the DTK41 10/100 port and the attached hub, switch or router will almost certainly result in unpredictable data error or loss.

5.34 ADMINISTER SECURITY BANNER

Syntax: banner [clear] [L#="Line # Message"]

The **banner** command is only visible when the unit is logged. It is used to administer the security banner. The default is a NULL banner. If a security banner is configured, it is displayed at each user login. The **clear** option is a shortcut to erase the entire message.

5.35 DATA-BASE RESET

Syntax: dbreset passwd=<password>

This command returns the **UMI** to the default configuration set up by the factory. The password will return to the factory default of ***initial***.

The **dbreset** command will always prompt for a password for validation purposes even if the administrator is logged at the appropriate level or higher.

6 UMI SNMP AGENT

The ***UMI*** SNMP V1 agent supports a multitude of SNMP MIB variables, SNMP ***trap*** operations, as well as ***set*** and ***get*** operations.

One or more SNMP managers may query the SNMP agent.

6.1 SNMP Version 1 Commands

Command	Operational Result
Get	Requests the values of one or more Management Information Base (MIB) variables.
GetNext	Enables MIB variables to be read sequentially, one variable at a time.
Set	Permits one or more MIB values to be updated.
GetResponse	Used to respond to a Get, GetNext, or Set.
Trap	Indicates the occurrence of a predefined condition.

6.2 UMI SNMP MIB Variable Database

RO = Read Only Variable
R/W = Read Variable / Write Variable
SIV = Storage is Volatile

MIB Variable Number	Name	MIB	Console Equivalent	Access	Notes
1.3.6.1.2.1.1.1.0	SysDescr	MIB-II	Banner Message	RO	
1.3.6.1.2.1.1.2.0	SysObjectID	MIB-II	None	RO	
1.3.6.1.2.1.1.3.0	SysUpTime	MIB-II	None	RO	
1.3.6.1.2.1.1.4.0	SysContact	MIB-II	None	R/W	SIV
1.3.6.1.2.1.1.5.0	SysName	MIB-II	None	R/W	SIV
1.3.6.1.2.1.1.6.0	SysLocation	MIB-II	None	R/W	SIV
1.3.6.1.2.1.1.7.0	SysServices	MIB-II	None	RO	
1.3.6.1.2.1.4.1.0	IpForwarding	MIB-II	None	RO	
1.3.6.1.2.1.4.2.0	IpDefaultTTL	MIB-II	None	RO	
1.3.6.1.2.1.4.3.0	IpInReceives	MIB-II	Nbr of Ethernet Pkts Rcvd	RO	
1.3.6.1.2.1.4.4.0	IpInHdrErrors	MIB-II	Nbr of Packets w/Header Errs	RO	
1.3.6.1.2.1.4.5.0	IpInAddrErrors	MIB-II	Nbr Rx Packets w/Wrong Addr	RO	
1.3.6.1.2.1.4.6.0	IpForwDatagrams	MIB-II	None	RO	
1.3.6.1.2.1.4.7.0	IpInUnknownProtos	MIB-II	Nbr of Packets w/Unk Protocol	RO	
1.3.6.1.2.1.4.8.0	IpInDiscards	MIB-II	Nbr of Packets Disc due to Resource	RO	

1.3.6.1.2.1.4.9.0	IpInDelivers	MIB-II	Inferred from DMEAS counters	RO	
1.3.6.1.2.1.4.10.0	IpOutRequests	MIB-II	Nbr of Device Frames Transmitted	RO	
1.3.6.1.2.1.4.11.0	IpOutDiscards	MIB-II	Nbr of Port frames Disc due to Resource	RO	
1.3.6.1.2.1.4.12.0	IpOutNoRoutes	MIB-II	None	RO	
1.3.6.1.2.1.4.13.0	IpReasmTimeout	MIB-II	None	RO	
1.3.6.1.2.1.4.14.0	IpReasmReqds	MIB-II	None	RO	
1.3.6.1.2.1.4.15.0	IpReasmOKs	MIB-II	None	RO	
1.3.6.1.2.1.4.16.0	IpReasmFails	MIB-II	None	RO	
1.3.6.1.2.1.4.17.0	IpFragOKs	MIB-II	None	RO	
1.3.6.1.2.1.4.18.0	IpFragFails	MIB-II	None	RO	
1.3.6.1.2.1.4.19.0	IpFragCreates	MIB-II	None	RO	
1.3.6.1.2.1.4.21.0	IpRoutingDiscards	MIB-II	None	RO	
1.3.6.1.2.1.5.1.0	IcmpInMsgs	MIB-II	None	RO	
1.3.6.1.2.1.5.2.0	IcmpInErrors	MIB-II	ICMP Errors	RO	
1.3.6.1.2.1.5.3.0	IcmpInDestUnreach	MIB-II	None	RO	
1.3.6.1.2.1.5.8.0	IcmpInEchos	MIB-II	Nbr of Pings	RO	
1.3.6.1.2.1.5.9.0	IcmpInEchoReps	MIB-II	None	RO	
1.3.6.1.2.1.6.1.0	TcpRtoAlgorithm	MIB-II	None	RO	
1.3.6.1.2.1.6.2.0	TcpRtoMin	MIB-II	None	RO	
1.3.6.1.2.1.6.3.0	TcpRtoMax	MIB-II	None	RO	
1.3.6.1.2.1.6.4.0	TcpMaxConn	MIB-II	None	RO	
1.3.6.1.2.1.6.5.0	TcpActiveOpens	MIB-II	None	RO	
1.3.6.1.2.1.6.6.0	TcpPassiveOpens	MIB-II	None	RO	
1.3.6.1.2.1.6.7.0	TcpAttemptFails	MIB-II	None	RO	
1.3.6.1.2.1.6.8.0	TcpEstabResets	MIB-II	None	RO	
1.3.6.1.2.1.6.9.0	TcpCurrEstab	MIB-II	None	RO	
1.3.6.1.2.1.6.10.0	TcpInSegs	MIB-II	None	RO	
1.3.6.1.2.1.6.11.0	TcpOutSegs	MIB-II	None	RO	
1.3.6.1.2.1.6.12.0	TcpRetransSegs	MIB-II	None	RO	
1.3.6.1.2.1.6.13.X	TcpConnTable Entries	MIB-II	None	RO	
1.3.6.1.2.1.6.14.0	TcpInErrs	MIB-II	None	RO	

1.3.6.1.2.1.6.15.0	TcpOutRsts	MIB-II	None	RO	
1.3.6.1.2.1.7.1.0	UdpInDatagrams	MIB-II	Derived from other Counts.	RO	
1.3.6.1.2.1.7.2.0	UdpNoPorts	MIB-II	Non-Peer and Spurious UDP errors	RO	
1.3.6.1.2.1.7.3.0	UdpInErrors	MIB-II	Frame Errors	RO	
1.3.6.1.2.1.7.4.0	UdpOutDatagrams	MIB-II	Frames Sent, Keep Alive Messages sent, etc.	RO	
1.3.6.1.2.1.7.5.X	udpEntry Table	MIB-II	None	RO	
1.3.6.1.2.1.11.1.0	SnmpInPkts	MIB-II	None	RO	
1.3.6.1.2.1.11.3.0	SnmpInBadVersions	MIB-II	None	RO	
1.3.6.1.2.1.11.4.0	SnmpInBadCommunityNames	MIB-II	None	RO	
1.3.6.1.2.1.11.5.0	SnmpInBadCommunityUses	MIB-II	None	RO	
1.3.6.1.2.1.11.6.0	SnmpInASNParseErrs	MIB-II	None	RO	
1.3.6.1.2.1.11.30.0	SnmpEnableAuthenTraps	MIB-II	None	R/W	SIV
1.3.6.1.2.1.11.31.0	SnmpSilentDrops	MIB-II	None	RO	
1.3.6.1.2.1.11.32.0	SnmpProxyDrops	MIB-II	None	RO	

7 SUPPORTED TRAPS

Alarm Text	Severity	Trap Type	Notes
None	N/A	ColdStart	Generated when the unit starts up
None	N/A	AuthFail	SNMP Authorization Failure

8 APPENDIX A: UMI MODULE MEASUREMENTS

This appendix itemizes the measurements available using the ***display measurements*** (**dm**) command with the **mod** option. These are module-based measurements. The base measurements are always displayed, while the error and exception counters are only displayed if nonzero.

Interface	Type	Protocol	Description
10BaseT	Base	All	Number of 10BaseT Packets Received
10BaseT	Base	All	Number of 10BaseT Packets Transmitted.
10BaseT	Except	All	Number of ICMP Echo Requests Received.
V.35, DSU, RS-232	Except	HDLC, FR	Number of Serial Trunk Frames Received.
V.35, DSU, RS-232	Except	HDLC, FR	Number of Serial Trunk Frames Transmitted.
DSU	Except	ATM	Number of ATM AAL5 Trunk Frames Received.
DSU	Except	ATM	Number of ATM AAL5 Trunk Frames Transmitted.
10BaseT	Error	All	Number of Ethernet Discards (Resource).
10BaseT	Error	All	Number of Late Collisions (TX).
10BaseT	Error	All	Number of Under-run. (TX).
10BaseT	Error	All	Number of packets which exceeded the Retry Limit (TX).
10BaseT	Error	All	Number of Carrier Sense Lost (TX).
10BaseT	Error	All	Number of Frame Collisions (RX).
10BaseT	Error	All	Number of Receiver Overruns (RX).
10BaseT	Error	All	Number of Receive CRC Errors. (RX).
10BaseT	Error	All	Number of Short Frame Errors. (RX).
10BaseT	Error	All	Number of Non-Aligned Frame Error. (RX).
10BaseT	Error	All	Number of Frame Length Violations. (RX).
10BaseT	Error	All	Number of Unsupported Protocol Frames. (RX).
10BaseT	Error	All	Number of Invalid UDP frames. (RX).
10BaseT	Error	All	Number of Rx Frames w/IP Header Checksum Errors. (RX).
10BaseT	Error	All	Number of Rx Frames w/ICMP Checksum Errors. (RX).
10BaseT	Error	All	Number of ICMP Unreachable Destination Messages (RX).
10BaseT	Error	IP-DSU	Number of Rx Frames from Non-Peer Entity.
10BaseT	Error	All	Number of Unknown ICMP Messages. (RX).
10BaseT	Error	IP-DSU	Number of Packets lost from TTL Network Error. (RX).
10BaseT	Error	All	Number of Packets with wrong IP Destination Address (RX).
10BaseT	Error	All	Number of Rx Packets with Unknown ARP Operations. (RX).
10BaseT	Error	All	Number of Bad ARP Reply Packets Received.
10BaseT	Error	All	Number of RFC894 Packets with an Unknown protocol type field. (RX).
10BaseT	Error	All	Number of 802.3 Frames with an Unknown protocol type field. (RX).
V.35, DSU RS-232	Error	HDLC, FR	Number of Frames aborted by CTS lost (TX).
V.35, DSU RS-232	Error	HDLC, FR	Number of Frames Under-Run. (TX).

V.35, DSU RS-232	Error	HDLC, FR	Number of Rx Frames Over-Run.
V.35, DSU RS-232	Error	HDLC, FR	Number of Rx Frames with CRC Errors.
V.35, DSU RS-232	Error	HDLC, FR	Number of Non-Aligned Frame Errors (RX).
V.35, DSU RS-232	Error	HDLC, FR	Number of Frame Length Violations. (RX).
DSU	Error	ATM	Number of ATM AAL5 Frames discarded due to buffer congestion.
V.35, DSU RS-232	Except	FR	Number of Frame Relay Frames received with the discard eligible (DE) bit set.
V.35, DSU RS-232	Except	FR	Number of Frame Relay Frames received with the Forward Explicit Congestion Notification (FECN) bit set.
V.35, DSU RS-232	Except	FR	Number of Frame Relay Frames received with the Backward Explicit Congestion Notification (BECN) bit set.
V.35, DSU RS-232	Except	FR	Number of Frame Relay Frames received with an Incorrect DLCI. These frames were therefore subsequently discarded.
10BaseT	Except	All	Number of SNMP Packets received outside CUG.

9 APPENDIX B: UMI VIRTUAL PORT MEASUREMENTS

This appendix itemizes the measurements available using the ***display measurements*** (**dm**) command with the **vport** option. These are the virtual-port-based measurements.

Interface	**Description**
URP, TCP	Number of Intervals with Ingress Data.
URP, TCP	Number of Intervals with Egress Data.
URP, TCP	Number of Intervals with Port errors.
URP Only.	Number of Intervals with URP Receiver Errors.
URP Only.	Number of Intervals with URP Re-transmissions.

Note: In the measurements above, an interval is defined as 3.2 seconds.

10 APPENDIX C: ALARMS

This appendix is an enumeration of the alarms presented on the ***UMI*** console along with their severity.

Alarm	Severity
Tx Error on 10BaseT. Check Physical Connection.	MAJOR
Insufficient Buffers for FRAME RELAY LMI Status Report.	MAJOR
User Requested Reboot in Progress	INFO
Invalid Login Attempt.	MINOR
Invalid Password Change Attempt.	MINOR
SNMP Trap Manager not reachable (ICMP).	INFO
ICMP Destination Unreachable Msg Received.	MINOR
Tx Error on Serial Interface	MAJOR
Over-Temperature Condition Detected.	MAJOR
Over-Temperature Condition Cleared.	INFO
High Temperature Condition Detected.	MINOR
High Temperature Condition Cleared.	INFO
Port XXX received a call from XXX.XXX.XXX.XXX outside CUG list.	MINOR
Serial Number is not valid. Module defective.	MAJOR
ATM Receiver never Synchronized.	INFO
Installation Attempt Failed.	MINOR
Console session in-activity timeout.	INFO
Vport XXX disconnected. Half Open TCP error.	INFO
DTK41 10/100 BaseT PHY #N 'Link Active'.	MINOR
DTK41 10/100 BaseT PHY #N 'Link InActive'.	MINOR
DTK41 Fiber 'Link Active'.	MINOR
DTK41 Fiber 'Link InActive'.	MINOR

10.1 Major Alarms

A major alarm indicates a serious, service-degrading condition.

10.2 Minor Alarms

A minor alarm indicates a secondary or transient error that is not likely to affect overall service unless multiple minor alarms are issued. In this case, a serious condition exists that may affect overall system performance

10.3 Info Alarms

An information alarm is a message that does not necessarily require attention. It typically is important for network administration, but does not adversely affect service.

11 APPENDIX D: USING STARKEEPER TO CONFIGURE THE UMI

Configuring a ***UMI*** in a BNS node consists of several steps. Since the ***UMI*** is represented as a SAM504 in the node, configuration information on the node must match the configuration information in the ***UMI*** in order for the module to operate correctly.

StarKeeper II NMS R10 provides a package, the ***UMI Configurator***, which supports the initial configuration of a ***UMI*** in a BNS node. This package performs these activities:

- Prompts the administrator for ***UMI*** virtual port configuration information;
- Performs validation on the entered information;
- Creates update specifications for both the ***UMI*** and BNS node configuration updates;
- Performs configuration update operations on both the ***UMI*** and BNS node.

This package does not perform the following configuration activities:

- Delete a currently configured BNS slot
- Process ***UMI*** Closed User Group (CUG) information
- Perform configuration updates to the BNS node after the initial configuration is established

NOTE: The use of ***UMI Configurator*** is optional. Administrators may use the appropriate module and BNS node commands to directly configure ***UMI*** units and BNS nodes.

11.1 PRE-CONFIGURATION ACTIVITIES

Several activities must take place before ***UMI*** and BNS configuration can take place.

11.1.1 Package Installation

Install the package on StarKeeper according to the instructions provided in the document *StarKeeper II NMS R10 Console Support for Datatek Products*. Establish a console connection from StarKeeper to the ***UMI*** console port, using the instructions provided in that document.

11.1.2 Develop Configuration Information

Develop the configuration information by planning how the ***UMI*** will be used in the network. The following worksheets can be used to list out the information needed to perform the configuration:

For ***UMI*** originating ports (BNS-to-IP call direction)

Node	Slot#	Start vport#	# vports	IP address (for PDD only)	dport (for PDD only)	BNS receive group name

For ***UMI*** receive ports (IP-to-BNS call direction)

Node	Slot#	Start vport#	# vports	hport# (optional)	BNS originating group name	PDD address (optional)

11.1.3 Perform BNS Activities

Perform these activities on the BNS node, using node commands:

- Check that the ***UMI*** slot is not configured. If it is, delete the configuration data.
- Enter the group names.
- Enter the service addresses.

11.1.4 Perform UMI Activities

Perform these activities on StarKeeper:

- Check that the console connection to the ***UMI*** is active.
- Check that the ***UMI*** is logged in

11.2 USING THE UMI CONFIGURATOR

The ***UMI Configurator*** is invoked in the following manner: In the **umi** directory created during script installation on StarKeeper, run command **umicfg.ctl**. The command works by providing a menu of operations from which the user selects an operation to be performed.

11.2.1 UMI Configurator Menu of Operations

When the administrator invokes the ***UMI Configurator***, the screen prompts appear as follows:

```
Enter node name: NNNNN
Enter node slot number: NN


1)  Display configuration data
2)  Create/Modify configuration data
3)  Update UMI with configuration data
4)  Update node with configuration data
5)  Exit processing
Select one number of the above:  N
```

The ‘node name’ and ‘slot number’ prompts identify the node and slot of the ***UMI***.

Menu item 1 allows the input configuration data to be displayed. This includes the administrator input, as well as the generated ***UMI*** and node update data.

Menu item 2 allows configuration data to be added and modified. Data is modified by deleting the previous entry and entering new input.

Menu item 3 causes the ***UMI*** to be configured.

Menu item 4 causes the BNS node to be configured.

Menu item 5 exits the ***UMI Configurator***.

11.2.2 Running the UMI Configurator

The ***UMI Configurator*** is used as follows:

1. Log onto the StarKeeper as user *cnmsadm.*
2. Verify that the StarKeeper console connection to the BNS node is active.
3. Verify that the console connection to the ***UMI*** is active.
4. Verify that the ***UMI*** console connection is logged in.
5. Change directory to directory **umi.**
6. Invoke the configuration script using the command **./umicfg.ctl**.
7. Create a new ***UMI*** configuration file by selecting the appropriate menu item..
8. Display and edit the configuration file, as appropriate.
9. Update the ***UMI*** with the configuration data.
10. Update the BNS node with the SAM configuration data. As the update takes place, messages will appear on the screen.
11. Exit the ***UMI Configurator***.

11.3 POST-CONFIGURATION ACTIVITIES

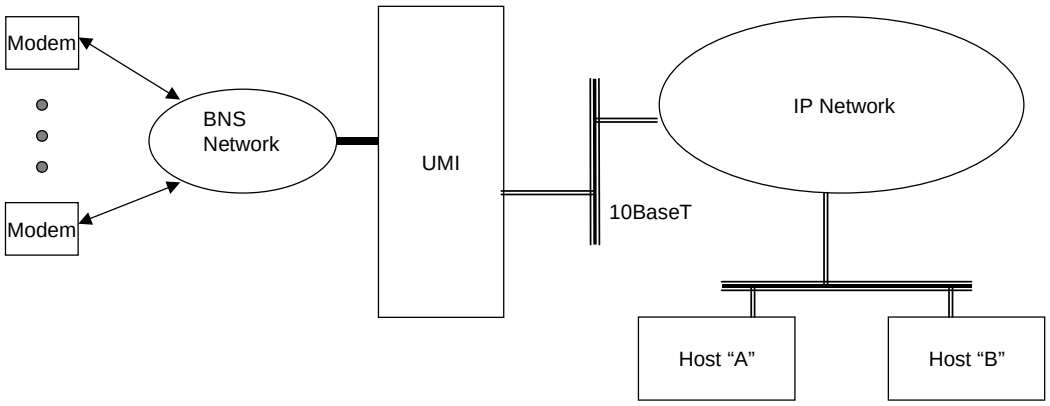
The ***UMI Configurator*** establishes an initial, consistent configuration in the ***UMI*** and BNS node. However, there are other activities that the administrator must perform. These include:

1. Restore the SAM to service. The ***UMI Configurator*** sets the ports in-service, but it leaves the boards and module out of service.
2. Define and add CUGs to virtual ports, if appropriate. This is done using the **cug** and **vport** module commands.
3. If the initial port configuration must be changed for any reason, use the appropriate node and ***UMI*** commands to make the changes.

12 APPENDIX E: UMI HUNT GROUP DEMONSTRATION

A ***UMI*** Hunt Group is a set of virtual ports arranged to receive calls at a common TCP port number. These calls may in turn be directed to a Predefined Destination (PDD) in the BNS network. Consider the following diagram:

UMI Hunt Group Demonstration



Suppose both **Host A** and **Host B** need to share a common BNS resource, such as a bank of modems. If each modem were individually addressed (with a PDD), the hosts would need to search through a list of ***UMI*** virtual ports (i.e., TCP ports), and attempt to call each one until a connection is successfully established. Since TCP connection timeouts are rather large, this would probably result in unacceptable performance.

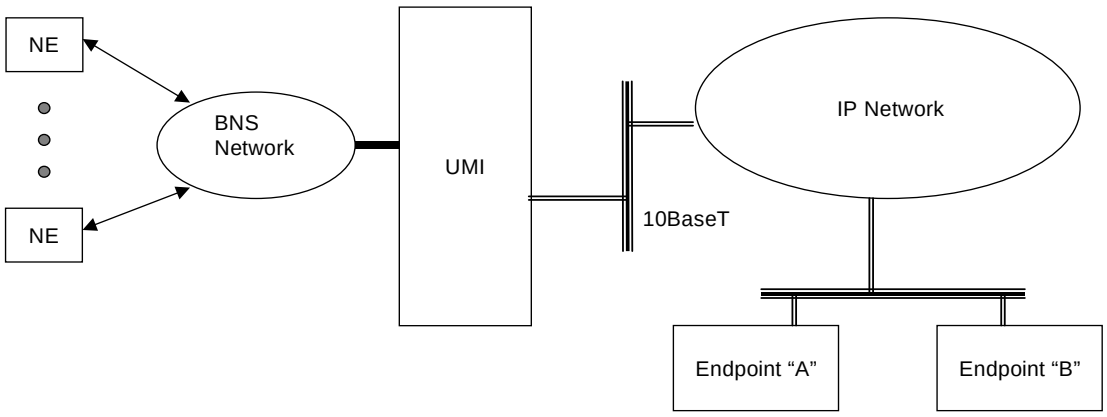
This is solved with the ***UMI*** by assigning a common TCP port number to a hunt group of virtual ports from which the modems are automatically dialed in the BNS network. Multiple distinct banks of modems may be dialed in the BNS network from a single ***UMI*** hunt group. The command sequence to configure the ***UMI*** for the desired operation was shown in section 4.3.2.

When a host calls the correct IP address and TCP port, it is connected to the next available virtual port in the hunt group. (A TCP connection timeout will take place only if no virtual ports are available.) When this connection is made, a BNS PDD directs the call to the address specified (by node administration) for the corresponding SAM504 port. This destination may actually be a BNS hunt group of ports, hence the ability to have a ***UMI*** hunt group act like a group of hunt groups.

13 APPENDIX F: UMI CLOSED USER GROUP DEMONSTRATION

The ***UMI*** supports the notion of ***Closed User Groups (CUGs)***. In a BNS network, CUGs are administered on the affected nodes, in order to restrict ***UMI*** virtual ports (i.e., SAM504 ports) to specific BNS endpoints, and vice versa. CUGs may also be fully extended from a ***UMI*** virtual port into the IP network. This can apply in either call direction (IP to BNS or BNS to IP), providing a secure firewall for BNS-IP connectivity. This is an important feature for protecting sensitive endpoints in a corporate-wide network without the burden of special "security servers"***.***

In the following diagram, there is a corporate IP network infrastructure which may be used by endpoints throughout the network. Some endpoints require access to Network Elements (NE) reachable via the BNS network, and some endpoints are not to be allowed such access. Those IP endpoints which are allowed access to the NE are placed in a CUG associated with a ***UMI*** virtual port. (The same CUG may be associated with any number of ***UMI*** virtual ports. Any one virtual port may have up to 128 CUGs.) A ***UMI*** virtual port is treated by the BNS Control Computer as a SAM504 port belonging to a BNS CUG, and the far endpoint in the BNS network also has a corresponding CUG association. Jointly, these CUG arrangements provide ***end-to-end*** security.



Referring to the above diagram, **Endpoint A** is allowed access to all the NEs. **Endpoint B** is not allowed access. Both are allowed access to the BNS network in general.

The ***UMI*** is configured with CUG 1 with the address of **Endpoint A**, as follows:

cug 1 ipaddr=135.17.59.5 submask=255.255.255.255

The ***protected*** virtual ports (i.e., those forming a hunt group which will be given access to the NEs via a separate node-administered BNS CUG) are set up with CUG 1 assigned to them, as follows:

vport 1 cnt=4 type=rcv hport=26 cug=+1

When **Endpoint A** calls the ***UMI*** and TCP port number 26, access to the NE in the BNS network is granted and everything proceeds transparently. If anyone outside CUG 1 (e.g., **Endpoint B**) attempts to call the same TCP port, however, the following happens:

1. The call is terminated during authentication without any data being transported in either direction.
2. An ***authentication alarm*** is generated and sent to an attached Starkeeper, an attached Telnet Console and the SNMP Trap Manager. The Alarm contains the IP address of the remote endpoint that attempted the unauthorized access.

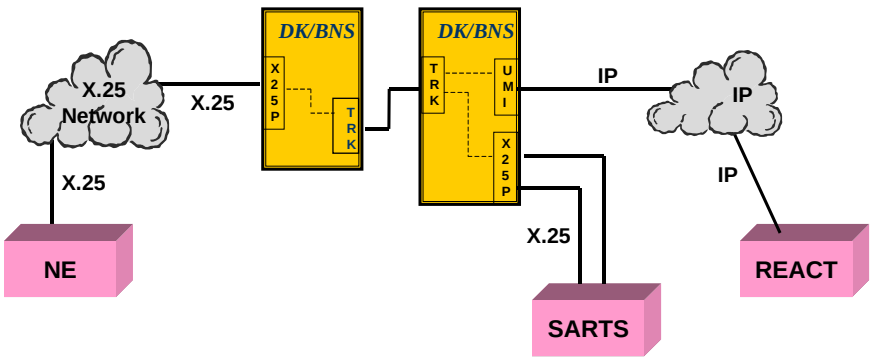
Endpoint B is given access to the BNS network in general by means of a separate hunt group on the ***UMI*** with a different TCP port number, which has no corresponding CUG restriction on the BNS network. This ***UMI*** hunt group might still have CUGs associated with it (for these two IP endpoints) if there are other endpoints in the IP network which should be ***totally*** restricted from accessing the BNS network.

14 APPENDIX G: VIRTUAL PAD EXAMPLE

In this example, an Datakit/BNS operations system; such as SARTS; is replaced with a native internet protocol operations systems. None of the element terminus points are changed. Further, both the old system and the new system can use the connections to the network elements.

VPAD Example

Extending the useful life of X25P Modules
REACT OS replaces SARTS OS



In the example above, the virtual ports used for REACT are set to a protocol of VPAD; and the requisite aggregation options are set on the virtual port.

The X.121 address used to address the X25P module in the DK/BNS has it's profile set to "transparent (min timer)".

When this occurs; the SARTS OS may continue to access the NE during the transition. The REACT OS may access the same NE via the IP interface and the VPAD.

The DK/BNS infrastructure is preserved during the OS transition.

For a full migration path to the NE that eliminates the DK/BNS network, please contact the author for a custom example using the DT-xx8x series of devices.

15 HARDWARE WARRANTY

The warranty period for the Universal Mediation Interface Module hardware shall be ninety (90) days from the date of shipment from TeleComp R&D or a designated manufacturer. Replacements and repairs are guaranteed for the longer of the remaining original warranty period or 30 days whichever is longer.

16 SOFTWARE END-USER LICENSE AGREEMENT

This License Agreement ("License") is a legal contract between you and the manufacturer ("Manufacturer") of the system ("HARDWARE") with which you acquired software product(s) identified above ("SOFTWARE"). The SOFTWARE may include printed materials that accompany the SOFTWARE. Any software provided along with the SOFTWARE that is associated with a separate end-user license agreement is licensed to you under the terms of that license agreement. By installing, copying, downloading, accessing or otherwise using the SOFTWARE, you agree to be bound by the terms of this LICENSE. If you do not agree to the terms of this LICENSE, Manufacturer is unwilling to license the SOFTWARE to you. In such event, you may not use or copy the SOFTWARE, and you should promptly contact Manufacturer for instructions on return of the unused product(s) for a refund.

16.1 Software License

You may only install and use one copy of the SOFTWARE on the HARDWARE (unless otherwise licensed by Manufacturer). The SOFTWARE may not be installed, accessed, displayed, run, shared or used concurrently on or from different computers, including a workstation, terminal or other digital electronic device (“Devices”). Notwithstanding the foregoing and except as otherwise provided below, any number of Devices may access or otherwise utilize the services of the SOFTWARE. You may not reverse engineer, decompile, or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation. The SOFTWARE is licensed as a single product. Its component parts may not be separated for use on more than one HARDWARE. The SOFTWARE is licensed with the HARDWARE as a single integrated product. The SOFTWARE may only be used with the HARDWARE as set forth in this LICENSE. You may not rent, lease or lend the SOFTWARE in any manner. You may permanently transfer all of your rights under this LICENSE only as part of a permanent sale or transfer of the HARDWARE, provided you retain no copies, you transfer all of the SOFTWARE (including all component parts, the media and printed materials, any upgrades, this LICENSE and, if applicable, the Certificate(s) of Authenticity), and the recipient agrees to the terms of this LICENSE. If the SOFTWARE is an upgrade, any transfer must also include all prior versions of the SOFTWARE. Without prejudice to any other rights, Manufacturer may terminate this LICENSE if you fail to comply with the terms and conditions of this LICENSE. In such event, you must destroy all copies of the SOFTWARE and all of its component parts.

16.2 Intellectual Property Rights

The SOFTWARE is licensed, not sold to you. The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. You may not copy the printed materials accompanying the SOFTWARE. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This LICENSE grants you no rights to use such content. All rights not expressly granted under this LICENSE are reserved Manufacturer and its licensors (if any).

16.3 Software Support

SOFTWARE support is not provided by Manufacturer, or its affiliates or subsidiaries separate from the HARDWARE. For SOFTWARE support, please contact your supplier of the HARDWARE. Should you have any questions concerning this LICENSE, or if you desire to contact Manufacturer for any other reason, please refer to the address provided in the documentation for the HARDWARE.

16.4 Export Restrictions

You agree that you will not export or re-export the SOFTWARE to any country, person, or entity subject to U.S. export restrictions. You specifically agree not to export or re-export the SOFTWARE: (i) to any country to which the U.S. has embargoed or restricted the export of goods or services, which as of March 1998 include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Sudan and Syria, or to any national of any such country, wherever located, who intends to transmit or transport the products back to such country; (ii) to any person or entity who you know or have reason to know will utilize the SOFTWARE or portion thereof in the design, development or production of nuclear, chemical or biological weapons; or (iii) to any person or entity who has been prohibited from participating in U.S. export transactions by any federal agency of the U.S. government.

16.5 Limited Warranty

Manufacturer warrants that (a) the SOFTWARE will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of shipment from TeleComp R&D or a designated manufacturer. Software support is limited to the hours of 9AM to 5PM ET Monday through Friday excluding TeleComp R&D observed holidays. An extended warranty may be purchased at additional cost. Any implied warranties on the SOFTWARE are limited to ninety (90) days. Some states/jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

Manufacturer's and its suppliers' entire liability and your exclusive remedy shall be, at Manufacturer's option, either (a) return of the price paid, or (b) repair or replacement of the SOFTWARE that does not meet this Limited Warranty and which is returned to Manufacturer with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

16.6 No Other Warranties

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, MANUFACTURER AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT, WITH REGARD TO THE SOFTWARE AND THE ACCOMPANYING WRITTEN MATERIALS. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

16.7 Limitation of Liability

To the maximum extent permitted by applicable law, in no event shall Manufacturer or its suppliers be liable for any damages whatsoever (including without limitation, special, incidental, consequential, or indirect damages for personal injury, loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use this product, even if Manufacturer has been advised of the possibility of such damages. In any case, Manufacturer's and its suppliers' entire liability under any provision of this License shall be limited to the amount actually paid by you for the SOFTWARE and/or the HARDWARE. Because some states/jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

16.8 Special Provisions

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and HARDWARE Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial HARDWARE Software-Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is TeleComp R&D or it's designee manufacturer., 102 SW Orange Blossom, Lake City, Florida, 32025-1613.

If you acquired the SOFTWARE in the United States of America, this Software License are governed by the laws of the State of Florida, excluding its choice of laws provisions. If you acquired the SOFTWARE outside the United States of America, local law may apply. This LICENSE constitutes the entire understanding and agreement between you and the Manufacturer in relation to the SOFTWARE and supercedes any and all prior or other communications, statements, documents, agreements or other information between the parties with respect to the subject matter hereof.

17 SALES & DISTRIBUTION

CBM of America, Inc.
Mr. Mike Stephens
1455 West Newport Center Drive
Deerfield Beach, Florida
33442

800-881-8202
954-698-9104 Fax: 954-360-0682

www.cbmusa.com

Datatek Applications, Inc.
Mr. Dan Conklin
379 Campus Drive, Suite 100
Somerset, New Jersey
08873

732-667-1080 Fax: 732-667-1091

www.datatekcorp.com

18 AUTHOR

Comments and Questions regarding this document or the products covered within this document should be addressed to the author Angel Gomez via email at angel@trdcusa.com or via telephone at 386-754-5700.

©Copyright 2002, 2009 TeleComp Research & Development Corp.
©Copyright 1998, 2002 TeleComp Inc.
All Rights Reserved
Printed in USA
Datakit® and StarKeeper® II NMS are registered trademarks of Lucent Technologies.